REPORT DOCUMENTATION PAGE

1. AGENCY USE ONLY (Leave Blank)

2. REPORT DATE
July 1997

3. REPO
Final (01 Oct 93 – 28 Feb 97)

4. TITLE AND SUBTITLE
Development of User-Matched/Network-Friendly Traffic Profiles for Congestion Control in High Speed Fiber Optics Networks

5. FUNDING NUMBERS
F49620-93-1-0564

6. AUTHORS
Professor Kurt Oughstun & Professor Ioannis Stavrakakis

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Vermont, Dept. of Electrical Engineering & Computer Science
Burlington, VT 05405

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NM
110 Duncan Avenue, Room B-115
Bolling Air Force Base, DC 20332-8080

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

In Wireless ATM networks transmission resources are shared among geographically disperse applications with disperse applications with diverse Quality of Service (QoS) requirements and traffic characteristics. To provide QoS guarantees and use the bandwidth efficiently, call admission and scheduling functions are necessary. These functions should ensure the delivery of the target QoS to the supported applications while achieving statistical multiplexing gains, without explicit and continuous exchange of information.

In this research the problem of sharing resources (slots of an up-link TDMA frame) among heterogeneous Variable Bit Rate (VBR) applications with diverse QoS requirements is addressed. The QoS requirements for each application is defined in terms of a maximum tolerable packet delay and dropping probability; a packet is dropped if it experiences excess delay. The region of achievable QoS vectors is established for policies that are work-conserving and satisfy the earliest due date (EDD) service criterion (WC-EDD policies); such policies are known to optimize the overall system performance. In addition to the determination of the region of achievable QoS vectors, this study leads also to the construction of scheduling policies which deliver any performance in the region established for WC-EDD policies. Finally, an upper bound on the region of QoS vectors that can be achieved under any policy (not limited to the WC-EDD policies) is determined.

14. SUBJECT TERMS
Quality of Service (QoS), Variable Bit Rate (VBR), vectors

)F PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

Accepted
9/21

FINAL TECHNICAL REPORT

Research supported by the Advanced Research Project Agency
(ARPA)
under Grant F49620-93-1-0564
Monitored by the Air Force Office of Scientific Research
(AFOSR)

Prof. Kurt Oughstun, *Acting Principal Investigator*
Department of Electrical Engineering and Computer Science
University of Vermont
Burlington, Vermont 05405
Tel: (802) 656-4301 / FAX (802) 656-0696 / Email: oughstun@emba.uvm.edu

Prof. Ioannis Stavrakakis, *Principal Investigator*
Department of Electrical & Computer Engineering
409 Dana Bldg.
Northeastern University
Boston, MA 02115.
Tel: (617) 373-3053 / Fax: (617) 373-8970 / Email: ioannis@cdsp.neu.edu

July 1997

FINAL TECHNICAL REPORT

Research supported by the Advanced Research Project Agency (ARPA)
under Grant F49620-93-1-0564
Monitored by the Air Force Office of Scientific Research (AFOSR)

The results of the research supported by this grant may be found in the listed publications. The abstracts provide for a summary of the research.

For completeness, all publications and presentations of the research to date are listed below but only the publications which were not submitted as part of the first, second or third year's report (September 1994 / October 1995 / September 1996) are attached in the Appendix. Research completed during the first, second or third year is indicated by referring to the First, Second or Third Interim Technical Report, respectively, for a copy of the related article. A copy of the publication of the research completed after the submission of the third year's report is included in the Appendix.

LIST OF ALL PUBLICATIONS / PRESENTATIONS OF THE RESEARCH

Journal Publications

1. M. Abdelaziz, I. Stavrakakis, "Some Optimal Traffic Regulation Schemes for ATM Networks: A Markov Decision Approach", *IEEE/ACM Transactions on Networking*, Vol. 2, No. 5, Oct. 1994.

   [See Second Interim Technical Report]

2. S. Tsakiridou, I. Stavrakakis, "Mean Delay Analysis of a Statistical Multiplexer with Batch Arrival Processes - A Generalization to Viterbi's Formula", *Performance Evaluation Journal*, Vol. 25, No. 1, March 1996.

   [See Third Interim Technical Report]

3. I. Stavrakakis, S. Tsakiridou, "Study of a Class of Partially Ordered Service Strategies for a System of Two Discrete-Time Queues", *Performance Evaluation Journal*, Vol. 29, pp. 15-33, 1997.

   [See Second Interim Technical Report]

   [A copy of the published version is included in the Appendix of this report]

4. R. Landry, I. Stavrakakis, "Multiplexing ATM Traffic Streams with Time-Scale-Dependent Arrival Processes", *Computer Networks and ISDN Systems*, Vol. 29, pp. 293-313, 1997.

   [See Second Interim Technical Report]

   [A copy of the published version is included in the Appendix of this report]

5. I. Stavrakakis, "Characterization and Multiplexing of Correlated Traffic from Sources with Different Time Constants", *Computer Communications Journal*, special issue on Enabling ATM Networks, Vol. 19, No. 13, Nov. 1996.

[See Third Interim Technical Report]

[A copy of the published version is included in the Appendix of this report]

6. J. Capone, I. Stavrakakis, "Achievable QoS and Scheduling Policies for Integrated Services Wireless Networks", *Performance Evaluation Journal*, to appear. [Also presented at the IFIP Performance'96 Conference, Oct. 7-11, 1996, Lausanne, Switzerland.]

   [See Third Interim Technical Report]

   [A copy of the published version is included in the Appendix of this report]

7. R. Landry, I. Stavrakakis, "Peak Rate Enforcement and its Impact on a Tagged ATM Session", *IEEE/ACM Transactions on Networking*, Vol. 5, No. 4, Aug. 1997.

   [See Second Interim Technical Report]

   [A copy of the (to be) published version is included in the Appendix of this report]

8. M. Abdelaziz, I. Stavrakakis, "Adaptive Rate Control in High-Speed Networks: Performance Issues", *Computer Networks and ISDN Systems*, to appear.

   [See Second Interim Technical Report]

9. T. Dag, I. Stavrakakis, "Evaluation of ABR Traffic Management Under Various System Time Scales", *Computer Networks and ISDN Systems*, to appear.

   [A copy of the article is included in the Appendix of this report]

Journal Submissions

10. I. Stavrakakis, G. Chen, "Study of a Scheduling Policy for Diversified Deadline-Based Quality-of-Service", *IEEE/ACM Transactions on Networking.*

    [See Second Interim Technical Report]

11. M. Conti, E. Gregori, I. Stavrakakis, "Impact of Temporal/Spatial Correlations on per Session Performance Measures: Single and Multiple Node case", *IEEE/ACM Transactions on Networking.*

    [See Third Interim Technical Report]

12. J. Capone, I. Stavrakakis, "Delivering QoS Requirements to Traffic with Diverse Delay Tolerances in a TDMA Environment", *IEEE/ACM Transactions n Networking.*

    [A copy of the article is included in the Appendix of this report]

13. I. Stavrakakis, S. Iatrou, "A Dynamic Regulation and Scheduling Scheme for Real Time Traffic Management", *IEEE/ACM Transactions on Networking.*

    [A copy of the article is included in the Appendix of this report]

Conference Proceedings

14. R. Landry, I. Stavrakakis, "Multiplexing Generalized Periodic Markovian Sources with an Application to the Study of VBR Video", proceedings of *International Conference on Communications (ICC'94)*, May 2-5, 1994, N. Orleans.

[See First Interim Technical Report]

15. R. Landry, I. Stavrakakis, "Non-Deterministic Periodic Packet Streams and Their Impact on a Finite-Capacity Multiplexer", proceedings of *IEEE INFOCOM'94*, June 12-16, 1994, Toronto, Canada.

    [See First Interim Technical Report]

16. R. Landry, I. Stavrakakis, "End-to-End Delay and Jitter Analysis in ATM Networks", proceedings of *2nd IFIP Workshop on Performance Modelling and Evaluation of ATM Networks*, July 4-6, 1994, University of Bradford, Bradford, UK.

    [See First Interim Technical Report]

17. M. Abdelaziz, I. Stavrakakis, "Open- and Closed-Loop Traffic Regulation Schemes for ATM Networks", proceedings of *IEEE GLOBECOM'94*, Nov. 27 - Dec. 1, 1994, San Francisco.

    [See First Interim Technical Report]

18. R. Landry, I. Stavrakakis, "A Queueing Study of Peak-Rate Enforcement for Jitter Reduction in ATM Networks", proceedings of *IEEE GLOBECOM'94*, Nov. 27 - Dec. 1, 1994, San Francisco.

    [See First Interim Technical Report]

19. R. Landry, I. Stavrakakis, "Traffic Shaping of a Tagged Stream in an ATM Network: Approximate End-to-End Analysis", proceedings of *IEEE INFOCOM'95*, April 2-6, Boston.

    [See Second Interim Technical Report]

20. M. Abdelaziz, I. Stavrakakis, "Study of an Adaptive Rate Control Scheme under Unequal Propagation Delays" *International Conference on Communications (ICC'95)*, June 18-22, 1995, Seattle.

    [See Second Interim Technical Report]

21. M. Conti, E. Gregori, I. Stavrakakis, "Study of the Impact of Temporal/Spatial Correlations on a Tagged Traffic Stream", 6^{th} *International Conference on Data Communication Systems and their Performance*, Oct. 23-26, 1995, Istanbul, Turkey. In *Data Communication Systems and their Performance*, ed.: S. Fdida, R. Onvural, Chapman & Hall, 1996.

    [See Second Interim Technical Report]

22. G. Chen, I. Stavrakakis, "Management of ATM Traffic with Diversified Loss and Delay Requirements", *IEEE INFOCOM'96*, March 26-28, 1996, San Francisco.

    [See Third Interim Technical Report]

23. X. Wang, I. Stavrakakis, "Study of Scheduling for Group-based Quality of Service Delivery *4th IFIP Workshop on Performance Modelling and Evaluation of ATM Networks*, July 8 - 10, 1996, West Yorkshire, U.K. Also, extended was published in *Third volume on Performance Modelling and Evaluation of ATM Networks*, ed.: D. Kouvatsos, Chapman & Hall, 1997.

    [See Third Interim Technical Report]

    [A copy of the extended version is included in the Appendix of this report]

24. S. Iatrou, I. Stavrakakis, "A Dynamic Real-Time Traffic Management Scheme with Good Throughput and Jitter Characteristics", *10th ITC Specialist Seminar in Control in Communications*, Sept. 17-19, 1996, Lund, Sweden.

    [See Third Interim Technical Report]

25. J. Capone, I. Stavrakakis, "Scheduling VBR Traffic for Achievable QoS in a Shared Unbuffered Environment", *International Conference on Universal Personal Communications (ICUPC'96)*, Sept. 29 - Oct. 2, 1996, Boston, MA.

    [See Third Interim Technical Report]

26. T. Dag, I. Stavrakakis, "Study of Impact of Network and Source Time Scales on Feedback-Based Congestion Control", *SPIE International Conference on Voice, Video, and Data Communications*, Nov. 18-22, 1996, Boston, MA.

    [See Third Interim Technical Report]

27. M. Abdelaziz, I. Stavrakakis, "Study of the Effectiveness of Adaptive Rate Control for Slow Arrival Processes", *IEEE GLOBECOM'96*, Nov. 18 - 22, 1996, London, U.K.

    [See Third Interim Technical Report]

28. T. Dag, I. Stavrakakis, "On the Impact of Time Scales on Feedback Based Congestion Control", *IEEE GLOBECOM'97*, Nov. 4-8, 1997, Phoenix, AZ.

    [A copy of the article is included in the Appendix of this report]

29. S. Iatrou, I. Stavrakakis, "An Integrated Regulation and Scheduling Scheme for Real-Time Traffic Management", *IEEE GLOBECOM'97*, Nov. 4-8, 1997, Phoenix, AZ.

    [A copy of the article is included in the Appendix of this report]

30. Z. Antoniou, I. Stavrakakis, "Earliest Due First Scheduling for Application-Level QoS Delivery", *IEEE Conference on Protocols for Multimedia Systems - Multimedia Networking (PROMS-MmNet'97)*, Nov. 24-26, Santiago, Chile.

    [A copy of the article is included in the Appendix of this report]

31. J. Capone, I. Stavrakakis, "Achievable QoS in a Shared Wireless Channel", *Fourth IEEE Workshop on the Architecture and Implementation of High Performance Communication Systems (HPCS'97)*, June 23-25, 1997, Sani Beach, Chalkidiki, Greece.

    [A copy of the article is included in the Appendix of this report]

Workshop Presentations

32. I. Stavrakakis, "Traffic Modeling and Regulation (Shaping) for High-Speed Networks", *Non-Linear Optics Communication Mountain Workshop*, April 12-13, 1994, Breckenridge, CO.

33. I. Stavrakakis, "Study of Some Rate Control and Buffer Management Schemes: Optimality and Fairness Issues", *First Workshop of the TC6 Task Force on Performance of Communication Systems*, Jan. 23-24, 1995, St. Thomas, Virgin Islands.

34. J. Capone, Ioannis Stavrakakis, "Delivering Diverse Delay/Dropping QoS Requirements in a TDMA Environment", *ITC Mini-Seminar on Performance Modeling and Design of Wireless/PCS Networks*, Oct. 3, 1996, Boston, MA.

35. I. Stavrakakis, J. Capone "On the Achievability of Diversified QoS Vectors" *Second Workshop of the IFIP TC6 WG6.3 on the Performance of Communication Systems*, Oct. 13-14, 1996, Balatonfured, Hungary.

APPENDIX

Publications not included in the past Reports

Reprinted from

PERFORMANCE EVALUATION

Performance Evaluation 29 (1997) 15–33

Study of a class of partially ordered service strategies for a system of two discrete-time queues *

Ioannis Stavrakakis [a,*], Sophia Tsakiridou [b]

[a] Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115, USA
[b] Department of Electrical and Computer Engineering, Queen's University, Kingston, Ont., Canada K7L 3N6

Received 2 February 1993; revised 20 November 1995

ELSEVIER

ELSEVIER

Performance Evaluation 29 (1997) 15-33

PERFORMANCE EVALUATION
An International Journal

Study of a class of partially ordered service strategies for a system of two discrete-time queues [★]

Ioannis Stavrakakis [a,*], Sophia Tsakiridou [b]

[a] *Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115, USA*
[b] *Department of Electrical and Computer Engineering, Queen's University, Kingston, Ont., Canada K7L 3N6*

Received 2 February 1993; revised 20 November 1995

Abstract

A discrete-time system of two queues served according to distinct policies is studied in this paper: one queue receives 1-limited service; the other is served according to some policy from the introduced class of (state dependent) policies $\mathcal{S}$. Class $\mathcal{S}$ contains traditional service policies as well as ad hoc policies. A partial performance ordering of policies in $\mathcal{S}$ is derived which can be useful in designing ad hoc policies with improved performance compared to that of traditional policies. Some examples of ad hoc policies in $\mathcal{S}$ are presented and numerical results are derived to illustrate the potential for efficiency provided by service policies in $\mathcal{S}$.

Keywords: Discrete-time queueing systems; M/G/1 type Markov chain; Matrix analytic technique; Service strategies

1. Introduction

Discrete-time queueing systems have widely been adopted for the study of packet communication networks, where packet processes are described in terms of discrete-time stochastic point processes. The service policies associated with such systems model the protocols that govern the availability of network resources to the packets, users or classes of users. The distributed nature of a networking system and/or the diversified service requirements of the supported users necessitate the development of sophisticated resource allocation protocols modeled in terms of queueing systems under various service policies. Examples are the exhaustive, gated and limited service policies which have been studied extensively, primarily in continuous time, in the context of polling [1,2], and vacation systems [3], and the consistent gated/limited policy [4] which has been applied for the study of the DQDB network [5,6].

[★] Research supported in part by the National Science Foundation under grant NCR-9011962 and by the Advanced Research Project Agency (ARPA) under Grant F49620-93-1-0564 monitored by the Air Force Office of Scientific Research (AFOSR). This work was conducted while the authors were with the Department of Computer Science and Electrical Engineering of the University of Vermont, Burlington, VT 05405, USA.

[*] Corresponding author. Tel.: (617) 373-3053; fax: (617) 373-8970.

PII S0166-5316(96)00002-8

A discrete-time single server queueing system consisting of two queues is considered in this paper. One of the queues receives 1-limited service, the other is served according to a policy contained in a versatile class of state-dependent service policies $\mathcal{S}$. This class contains some of the well-known service policies as well as new ad hoc policies which may provide for improved performance compared to that under traditional policies.

The system studied in this paper may be considered as a variant of a two-queue asymmetric polling system with 1-limited service at one queue and service according to a policy in $\mathcal{S}$ at the other. When both queues receive 1-limited service the system operates under an alternating service policy, which has been analyzed in continuous time [7]. A continuous-time two-queue model with mixed exhaustive and limited services has been studied in [8]. A discrete-time model with gated and 1-limited services has been analyzed in [4].

The system considered in this paper may also be described in terms of a GI/D/1 infinite queue with server vacations. The vacation period distribution depends on the occupancy of the queue which is served during the vacation. The queueing model and analysis developed in this paper are also applicable to a system with vacation period distributions that depend on the state of a Markov process which evolves independently of the system [9]. This state-dependent vacation model is different from most of those presented in the past, for example [10,11], in the following aspect: the process whose state determines the occurrence and distribution of vacation periods is external to the queue under study. The analysis is based on matrix analytic techniques similar to those applied for the study of the vacation model in [12].

The detailed description of the queueing system and a unified representation of the class of service policies $\mathcal{S}$ are presented in Section 2. In Section 3, a queueing model for the system is formulated and analyzed. The joint probability distribution of the queue occupancies is derived by applying matrix analytic methods and Markov renewal theory arguments. Section 4 presents a partial performance ordering for policies in $\mathcal{S}$. Some numerical results which illustrate the effect of various service disciplines on the performance of the system are presented in Section 5. The work is summarized in Section 6.

2. Description of the queueing system

2.1. *Introduction*

Consider the discrete-time queueing system shown in Fig. 1. The packet service time is assumed to be constant and equal to the system time unit (slot). Unless otherwise stated, a superscript I (F) will indicate a quantity associated with queue Q^{I} (Q^{F}). The capacities of queues Q^{F} and Q^{I} are assumed to be $N < \infty$ and infinite, respectively. Let $\{q_j^{\mathrm{I}}\}_{j\geq 0}$ and $\{q_j^{\mathrm{F}}\}_{j\geq 0}$ denote the associated queue occupancy processes with state spaces $R^{\mathrm{I}} = \{0, 1, 2, \ldots\}$ and $R^{\mathrm{F}} = \{0, 1, \ldots, N\}$.

The system service discipline is determined by the general rules presented in Section 2.2 and the adopted service policy in $\mathcal{S}$ presented in Section 2.3. A brief introduction to the service discipline is presented first.

The server never idles as long as there is service to be performed. Upon switching to Q^{F}, the server serves a number of F-packets (packets in Q^{F}) which depends on the number of packets found in Q^{F} and the number of packets left in Q^{I}. If the number of F-packets served cannot exceed the number of F-packets already present in Q^{F} at the switching instant, then this state-policy (see Section 2.3) will be called gated. Otherwise, it will be called non-gated. If all state-policies of a service policy are gated (non-gated) the service policy will also be called gated (non-gated). Following the service to Q^{F} the server switches to Q^{I}

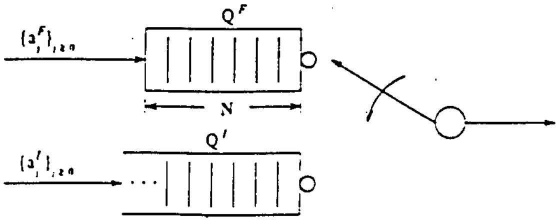


Fig. 1. The queueing system.

and provides 1-limited service to that queue. Although a service policy other than the 1-limited could be considered for Q^{I}, the analysis developed in this work is applicable only under 1-limited service to Q^{I}.

2.2. *The general service mechanism*

The system is work conserving; that is, the server is never idle when the system is non-empty. The server is assumed to be at Q^{F} when idle. Let a *decision time-instant* t^{d} be defined as the time instant at which a packet is forwarded to the head of Q^{I}; the collection of decision time-instants is denoted by $\mathcal{A}$.

(i) At t^{d}, $t^{\mathrm{d}} \in \mathcal{A}$, the server switches to Q^{F}; the amount of service provided to this queue is determined by the adopted policy in $\mathcal{S}$ (Section 2.3). Upon completion of this service to Q^{F}—which can possibly be equal to 0—the server switches to Q^{I} and serves one packet.

(ii) Upon completion of the 1-limited (packet) service to Q^{I}:

(a) If Q^{I} is left non-empty, a packet is instantaneously forwarded to the head of Q^{I} and, thus, a decision time-instant t^{d} is reached.

(b) If Q^{I} is left empty, Q^{F} is served uninterruptedly until the next decision time-instant t^{d}, to be determined by the first future packet arrival to Q^{I}.

Notice that before service is provided to a packet forwarded to the head of Q^{I}, an amount of service is provided to Q^{F}, as determined by a policy in $\mathcal{S}$. In general, the amount of service provided to Q^{F}—determined by the policy in $\mathcal{S}$—will depend on the state (queue occupancy) of Q^{F}. Thus, in general, the server behavior is 'regulated' by the state of Q^{F}. Since the server considers switching back to Q^{F} at any (discrete) time instant at which it is away from this queue, Q^{F} may be viewed as the critical queue. Although almost the entirety of the adopted service disciplines for Q^{F} would provide prioritized service to this queue, there is at least one such service discipline under which the priority is reversed.

2.3. *The class of service policies $\mathcal{S}$*

As will become clear, a number of traditional service disciplines, as well as ad hoc policies—potentially improving the performance achieved by traditional ones—can be represented by selecting a proper service discipline in $\mathcal{S}$.

Let i, $0 \leq i \leq N$, denote the state of $\{q_j^{\mathrm{F}}\}_{j\geq 0}$ at some decision time-instant t^{d}, $t^{\mathrm{d}} \in \mathcal{A}$. A *state-policy* associated with state i, $\mathcal{P}_i$, is defined to be equal to the *potential* amount of service (in packets) provided to Q^{F} by the server, immediately after t^{d}, $0 \leq \mathcal{P}_i \leq \infty$. This amount of service will be provided to Q^{F} unless the queue becomes empty. In view of rule (i), the definition of $\mathcal{P}_i$ and the work-conserving nature of the service discipline, the server will switch to Q^{I} at $t^{\mathrm{d}} + \tau(i)$, where $\tau(i) = \min\{\mathcal{P}_i, t^{\mathrm{F}}(i)\}$ and $t^{\mathrm{d}} + t^{\mathrm{F}}(i)$

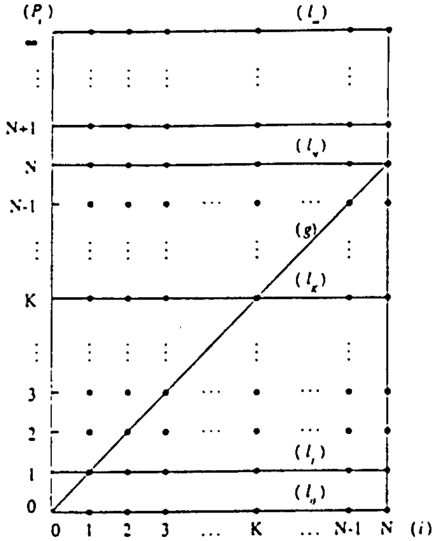


Fig. 2. The space of state-policies $\mathcal{SP}$ (each dot marks a state-policy point).

denotes the first time-instant following t^{d} at which Q^{F} becomes empty; $\tau(q^{\mathrm{F}}_{t^{\mathrm{d}}})$ will be referred to as the *service horizon* following t^{d} or determined by $q^{\mathrm{F}}_{t^{\mathrm{d}}}$.

A service policy in $\mathcal{S}$ will be defined in terms of the state-policies $\mathcal{P}_i$ associated with all states $i, 0 \leq i \leq N$. Since the service policies considered here are work-conserving, it is easily established that any selection for $\mathcal{P}_0$ will induce identical server behavior as $\mathcal{P}_0 = 0$. $\mathcal{P}_0 = 0$ will be assumed for any policy in $\mathcal{S}$ and, thus, a policy in $\mathcal{S}$ will be defined in terms of $\mathcal{P}_i$, $1 \leq i \leq N$. A service policy in $\mathcal{S}$ will be uniquely represented by the N-dimensional vector $\bar{\mathcal{P}} = (\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_N)$, $\bar{\mathcal{P}} \in \mathcal{S} = \mathbb{Z}_0^+ \times \mathbb{Z}_0^+ \times \cdots \times \mathbb{Z}_0^+$, where $\mathbb{Z}_0^+$ denotes the set of non-negative integers. As will become clear below, this vector representation of a service policy in terms of the state-policies $\{\mathcal{P}_i\}_{i=1}^{N}$ not only provides for a unified description (and study) of many well known service policies, but also allows for the introduction of new ad hoc policies. The space of state-policies $\mathcal{SP}$ is shown in Fig. 2. A service policy $\bar{\mathcal{P}} \in \mathcal{S}$ may be graphically represented by the set of *state-policy points* $\{(i, \mathcal{P})\colon 1 \leq i \leq N\}$, or the lines connecting these points (*policy-lines*) on $\mathcal{SP}$.

The following definitions will provide for a classification of state-policies and service policies in $\mathcal{S}$.

Definition 1. A state-policy $\mathcal{P}_i$ will be called *gated* if only packets that are present in Q^{F} at some $t^{\mathrm{d}} \in \mathcal{A}$ at which $q^{\mathrm{F}}_{t^{\mathrm{d}}} = i$ will be served over the service horizon following t^{d}; otherwise, it will be called *non-gated*. It will be called *unlimited (limited) gated* if all (not all) packets that are present in Q^{F} at some $t^{\mathrm{d}} \in \mathcal{A}$ at which $q^{\mathrm{F}}_{t^{\mathrm{d}}} = i$ will be served over the service horizon following t^{d}. It will be called *unlimited non-gated* if exhaustive service is provided to Q^F; otherwise, it will be called *limited non-gated*.

Definition 2. A service policy $\bar{\mathcal{P}} = (\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_N) \in \mathcal{S}$ will be called (*limited* or *unlimited*) *gated* if all associated state-policies are (limited or unlimited) gated. It will be called (*limited* or *unlimited*) *non-gated* if all associated state-policies are (limited or unlimited) non-gated.

Based on Definition 1, it can be easily established that $\mathcal{P}_i$ is gated if $\mathcal{P}_i \leq i$. In this case, $\mathcal{P}_i$ packets will be served over the service horizon $\tau(i)$, and thus, $\Pr\{\tau(i) = \mathcal{P}_i\} = 1$. $\mathcal{P}_i$ is unlimited gated if $\mathcal{P}_i = i$ and limited gated if $\mathcal{P}_i < i$. Similarly, $\mathcal{P}_i$ is non-gated if $\mathcal{P}_i > i$. In this case, *at least* i packets will be served over the service horizon $\tau(i)$ and $\Pr\{\tau(i) = \mathcal{P}_i\} < 1$. $\mathcal{P}_i$ is unlimited non-gated if $\mathcal{P}_i = \infty$ and limited non-gated if $\mathcal{P}_i < \infty$.

If $\bar{\mathcal{P}}$ is gated, the amount of service provided to Q^{F} over any service horizon $\tau(i)$ is equal to $\mathcal{P}_i$, for all i, $1 \leq i \leq N$, thus: $\Pr\{\tau(i) = \mathcal{P}_i\} = 1$. The subclass of gated service policies $\mathcal{S}_{\mathrm{G}}$ is defined on the subspace $\mathcal{SP}_{\mathrm{G}} = \{(i, \mathcal{P}_i): 0 \leq \mathcal{P}_i \leq i,\ 0 \leq i \leq N\}$ of $\mathcal{SP}$.

If $\bar{\mathcal{P}}$ is non-gated, the amount of service provided to Q^{F} over a service horizon $\tau(i)$ is at least equal to $\mathcal{P}_i$ and depends on the evolution of $\{q_j^{\mathrm{F}}\}_{j \geq 0}$ over that service horizon; thus, $\Pr\{\tau(i) = \mathcal{P}_i\} < 1$. The subclass of non-gated service policies $\mathcal{S}_{\mathrm{NG}}$ is defined on the subspace $\mathcal{SP}_{\mathrm{NG}} = \{(i, \mathcal{P}_i): \mathcal{P}_i > i, 0 \leq i \leq N\}$ of $\mathcal{SP}$.

Based on the above definitions, the space of state-policies $\mathcal{SP}$ can be divided into the subspaces of gated, $\mathcal{SP}_{\mathrm{G}}$, and non-gated, $\mathcal{SP}_{\mathrm{NG}}$, state-policies; a policy-line in $\mathcal{SP}_{\mathrm{G}}$ ($\mathcal{SP}_{\mathrm{NG}}$) defines a gated (non-gated) service policy in $\mathcal{S}$. The boundary between the two subspaces identifies the diagonal policy-line (g) (Fig. 2) which represents the unlimited gated service policy. The following definition provides for another classification of service policies in $\mathcal{S}$.

Definition 3. A service policy $\bar{\mathcal{P}}$ is k-limited, if $\mathcal{P}_i = k$, $1 \leq i \leq N$, for some k, $0 \leq k \leq \infty$. Note that a k-limited service policy is not state-dependent. This is also indicated by the graphical representation of the k-limited service policies in $\mathcal{SP}$; each k-limited service policy, $0 \leq k \leq \infty$, is represented by a horizontal policy-line (l_k) (Fig. 2).

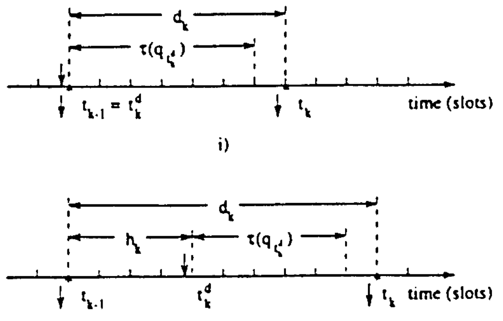
The state-policies $\mathcal{P}_i$ associated with a k-limited service policy $\bar{\mathcal{P}}$ may be gated ($i \geq k$) or non-gated ($i < k$). For the extreme values of k, $k = 0$ and $k = \infty$, the system service policy becomes HoL priority for queues Q^{I} and Q^{F}, respectively. For $k = 1$, the system operates under an alternating service policy: a single packet is served at each queue, in a alternating manner, while both queues are non-empty.

2.4. An expansion of class $\mathcal{S}$, $\mathcal{S}_{\mathrm{exp}}$

Based on the definition of a state-policy stated at the beginning of Section 2.3, the potential amount of service over any service horizon $\tau(i)$, for any policy $\bar{\mathcal{P}} \in \mathcal{S}$ described so far, is deterministic and equal to $\mathcal{P}_i$. A generalized definition of a state-policy is introduced below to describe probabilistically limited service policies [13] and expand the class of service policies in $\mathcal{S}$. A probabilistically limited state-policy associated with state i, $1 \leq i \leq N$, may be defined in terms of the probability vector $\bar{g}_i = [g_{i0}\ g_{i1}\ g_{i2} \ldots]^{\mathrm{T}}$, where g_{ij}, $1 \leq i \leq N$, $j \geq 0$, denotes the probability that the *potential* amount of service provided to Q^{F} over a service horizon $\tau(i)$ is equal to j. A service policy in the expanded class of service policies $\mathcal{S}_{\mathrm{exp}}$ may be represented by a matrix $\bar{\mathcal{P}}$: $\bar{\mathcal{P}} = [\bar{g}_1\ \bar{g}_2\ \bar{g}_3 \ldots \bar{g}_N]$.

Clearly, $\mathcal{S} \subset \mathcal{S}_{\mathrm{exp}}$. A service policy $\bar{\mathcal{P}} \in \mathcal{S}$ can be expressed in terms of a matrix $\bar{\mathcal{P}}$ by assigning the proper distributions to every (deterministic) state-policy i:

$$g_{ij} = \begin{cases} 1 & \text{for } j = \mathcal{P}_i, \\ 0 & \text{otherwise}, \end{cases} \qquad 1 \leq i \leq N.$$



i)

ii)

Fig. 3. Illustration of the definitions h_k and d_k: (i) $h_k = 0$, (ii) $h_k > 0$. An arrow pointing to (away from) the time axis indicates the movement of a packet to (away from) the head of Q^{I}.

A probabilistically limited service policy is gated if $\sum_{j=0}^{i} g_{ij} = 1$ and non-gated if $\sum_{j=i+1}^{\infty} g_{ij} = 1$, $1 \leq i \leq N$. Examples of probabilistically limited service policies are the Bernoulli gated:

$$g_{ij} = \begin{cases} 1 - p_i, & j = 0, \\ p_i, & j = 1, \\ 0, & \text{otherwise}, \end{cases} \qquad 1 \leq i \leq N,$$

and the binomial gated:

$$g_{ij} = \begin{cases} \binom{i}{j} p_i^j (1 - p_i)^{i-j}, & 0 \leq j \leq i, \\ 0, & \text{otherwise}, \end{cases} \qquad 1 \leq i \leq N,$$

where $0 \leq p_i \leq 1$, $1 \leq i \leq N$, and $\binom{i}{j}$ is the binomial coefficient.

For $p_1 = p_2 = \cdots = p_N = 1$ the Bernoulli gated policy becomes 1-limited and the binomial gated service policy becomes unlimited gated. For $p_1 = p_2 = \cdots = p_N = 0$ both policies become HoL priority at Q^{I}. In the rest of this paper, only service policies in $\mathcal{S}$ will be considered.

3. Modeling and analysis of the queueing system

The queueing behavior of the system under the adopted service discipline is analyzed in this section under i.i.d. and mutually independent arrival processes $\{a_j^{\mathrm{F}}\}_{j\geq 0}$ and $\{a_j^{\mathrm{I}}\}_{j\geq 0}$ to Q^{F} and Q^{I}, respectively. The joint probability distribution of the occupancy at queues Q^{I} and Q^{F} upon service completion of a packet at Q^{I} is derived first, based on a matrix analytic approach. The joint probability distribution of the queue occupancies at arbitrary time instants is then obtained by applying Markov renewal theory arguments. Consider the following time sequences defined at the end of slots:

$\{t_k\}_{k\geq 0}$: Defined to be the sequence of time instants at which the service of a packet is completed at Q^{I} ($t_0 = 0$).

$\{t_k^{\mathrm{d}}\}_{k\geq 1}$: Defined to be the sequence of decision time-instants, or time instants at which a packet is forwarded to the head of Q^{I} ($\{t_k^{\mathrm{d}}\}_{k\geq 1} \equiv \mathcal{A}$).

According to the system service discipline rules described in Section 2.2, the server switches to Q^{F} at t_{k-1} for an amount of time which depends on $q^{\mathrm{I}}_{t_{k-1}}$ and $q^{\mathrm{F}}_{t^{\mathrm{d}}_k}$:

(i) If $q^{\mathrm{I}}_{t_{k-1}} > 0$, by rule (ii(a)), the kth packet is instantaneously forwarded to the head of Q^{I}, and thus, $t_{k-1} = t^{\mathrm{d}}_k$ (Fig. 3(i)). In this case, the server will remain at Q^{F} for $\tau(q^{\mathrm{F}}_{t^{\mathrm{d}}_k})$ slots.

(ii) If $q^{\mathrm{I}}_{t_{k-1}} = 0$, by rule (ii(b)), Q^{I} will remain empty for a time interval h_k, until the first future packet arrival to Q^{I}; at $t_{k-1} + h_k$ the decision time-instant t^{d}_k is reached (Fig. 3(ii)). In this case, the server will remain at Q^{F} for $h_k + \tau(q^{\mathrm{F}}_{t^{\mathrm{d}}_k})$ slots.

The service of the kth packet at Q^{I} will be completed at time instant $t_k = t^{\mathrm{d}}_k + \tau(q^{\mathrm{F}}_{t^{\mathrm{d}}_k}) + 1$ and, therefore, the time interval d_k between the successive service completions of the $(k-1)$st and kth packets at Q^{I} is given by $d_k = t_{k-1} - t_k = h_k + \tau(q^{\mathrm{F}}_{t^{\mathrm{d}}_k}) + 1$ where $h_k = 0$, for $q^{\mathrm{I}}_{t_{k-1}} > 0$ (Fig. 3). It will be shown in the following that the process $\{q^{\mathrm{I}}_{t_k}, q^{\mathrm{F}}_{t_k}\}_{k\geq 0}$ is a Markov chain embedded at service completion time instants $\{t_k\}_{k\geq 0}$. In order to proceed with the analysis the following random variables need to be defined:

a^{I} (a^{F}) : Denotes the number of packet arrivals to Q^{I} (Q^{F}) in an arbitrary slot. The probability distribution of a^{I} (a^{F}) is denoted by $p^{\mathrm{I}}_a(n)$ $(p^{\mathrm{F}}_a(n))$, $n \geq 0$.

$a^{\mathrm{I},l}$: Denotes the number of packet arrivals to Q^{I} over l slots. It is distributed according to $p^{\mathrm{I},l}_a(n)$, the l-fold convolution of $p^{\mathrm{I}}_a(n)$, $n \geq 0$ $(p^{\mathrm{I},1}(n) \equiv p^{\mathrm{I}}_a(n))$.

h : Denotes the generic random variable for h_k; it describes the time between the service completion of a packet that leaves Q^{I} empty and the first packet arrival to the empty queue. The distribution of random variable h is given by $p_h(n) = [p^{\mathrm{I}}_a(0)]^{n-1}[1 - p^{\mathrm{I}}_a(0)]$, $n \geq 1$.

Proposition 1. *The process* $\{q^{\mathrm{I}}_{t_k}, q^{\mathrm{F}}_{t_k}\}_{k\geq 0}$ *is a two-dimensional Markov chain embedded on packet service completion time instants,* $\{t_k\}_{k\geq 0}$.

Proof. It suffices to show that $q^{\mathrm{I}}_{t_k}$ and $q^{\mathrm{F}}_{t_k}$ can probabilistically be determined from $q^{\mathrm{I}}_{t_{k-1}}$ and $q^{\mathrm{F}}_{t_{k-1}}$. The evolution of the occupancy process $\{q^{\mathrm{F}}_j\}_{j\geq 0}$ is determined based on the server availability to Q^{F}; transitions take place at the end of slots. If the server is at Q^{F} at the end of a slot, then process $\{q^{\mathrm{F}}_j\}_{j\geq 0}$ makes a transition according to the probability matrix $\boldsymbol{P_s}$,

$$\boldsymbol{P_s} = \begin{bmatrix} p^{\mathrm{F}}_a(0) & p^{\mathrm{F}}_a(1) & p^{\mathrm{F}}_a(2) & \cdots & p^{\mathrm{F}}_a(N-1) & \sum\limits_{n=N}^{\infty} p^{\mathrm{F}}_a(n) \\ p^{\mathrm{F}}_a(0) & p^{\mathrm{F}}_a(1) & p^{\mathrm{F}}_a(2) & \cdots & p^{\mathrm{F}}_a(N-1) & \sum\limits_{n=N}^{\infty} p^{\mathrm{F}}_a(n) \\ 0 & p^{\mathrm{F}}_a(0) & p^{\mathrm{F}}_a(1) & \cdots & p^{\mathrm{F}}_a(N-2) & \sum\limits_{n=N-1}^{\infty} p^{\mathrm{F}}_a(n) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & p^{\mathrm{F}}_a(0) & \sum\limits_{n=1}^{\infty} p^{\mathrm{F}}_a(n) \end{bmatrix}. \qquad (1)$$

If the server is absent from Q^{F} at the end of a slot, that is, at time instants $\{t_k\}_{k\geq 0}$ transitions take place according to the probability matrix P_v.

$$P_v = \begin{bmatrix} p_a^{\mathrm{F}}(0) & p_a^{\mathrm{F}}(1) & p_a^{\mathrm{F}}(2) & \cdots & p_a^{\mathrm{F}}(N-1) & \sum\limits_{n=N}^{\infty} p_a^{\mathrm{F}}(n) \\ 0 & p_a^{\mathrm{F}}(0) & p_a^{\mathrm{F}}(1) & \cdots & p_a^{\mathrm{F}}(N-2) & \sum\limits_{n=N-1}^{\infty} p_a^{\mathrm{F}}(n) \\ 0 & 0 & p_a^{\mathrm{F}}(0) & \cdots & p_a^{\mathrm{F}}(N-3) & \sum\limits_{n=N-2}^{\infty} p_a^{\mathrm{F}}(n) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix}. \qquad (2)$$

Therefore, the l-step transition probability $\Pr\{q_{j+l}^{\mathrm{F}} = j \mid q_j^{\mathrm{F}} = i\}$ of the process over an interval $(j, j+l]$ of uninterrupted service to Q^{F} of length l (slots) is given by $P_q^{\mathrm{F}}(l) = [P_s]^l$, $l \geq 1$. Given $(q_{t_{k-1}}^{\mathrm{I}}, q_{t_{k-1}}^{\mathrm{F}})$, $q_{t_k^{\mathrm{d}}}^{\mathrm{F}}$ is determined by $q_{t_{k-1}}^{\mathrm{F}}$ and the transition matrix $P_q^{\mathrm{F}}(h_k)$, where $h_k = 0$, if $q_{t_{k-1}}^{\mathrm{I}} > 0$ and distributed as random variable h otherwise; since $t_k = t_k^{\mathrm{d}} + \tau(q_{t_k^{\mathrm{d}}}^{\mathrm{F}}) + 1$, $q_{t_k}^{\mathrm{F}}$ is determined by $q_{t_k^{\mathrm{d}}}^{\mathrm{F}}$ and the transition matrix $P_q^{\mathrm{F}}(\tau(q_{t_k^{\mathrm{d}}}^{\mathrm{F}}))P_v$. Since $q_{t_k}^{\mathrm{I}} = q_{t_{k-1}}^{\mathrm{I}} + a^{\mathrm{I},d_k} - 1$ and $d_k = h_k + \tau(q_{t_k^{\mathrm{d}}}^{\mathrm{F}}) + 1$, $q_{t_k}^{\mathrm{I}}$ is determined by $(q_{t_{k-1}}^{\mathrm{I}}, q_{t_{k-1}}^{\mathrm{F}})$, in view of the above discussion. □

By identifying $q_{t_k}^{\mathrm{I}}$ and $q_{t_k}^{\mathrm{F}}$ as the level and phase of process $\{q_{t_k}^{\mathrm{I}}, q_{t_k}^{\mathrm{F}}\}_{k\geq 0}$, respectively, the transition probability matrix $P_{(q^{\mathrm{I}},q^{\mathrm{F}})}$ of this embedded Markov chain can be written as a stochastic matrix of M/G/1 type:

$$P_{(q^{\mathrm{I}},q^{\mathrm{F}})} = \begin{bmatrix} B_0 & B_1 & B_2 & B_3 & B_4 & \cdots \\ A_0 & A_1 & A_2 & A_3 & A_4 & \cdots \\ 0 & A_0 & A_1 & A_2 & A_3 & \cdots \\ 0 & 0 & A_0 & A_1 & A_2 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \qquad (3)$$

where A_n and B_n, $n \geq 0$, are the $(N+1)$-dimensional matrices with elements $A_n(j_1, j_2)$ and $B_n(j_1, j_2)$, $j_1, j_2 \in R^{\mathrm{F}}$, given by

$$A_n(j_1, j_2) = \Pr\left\{q_{t_k}^{\mathrm{I}} = i_1 - 1 + n, q_{t_k}^{\mathrm{F}} = j_2 \mid q_{t_{k-1}}^{\mathrm{I}} = i_1, q_{t_{k-1}}^{\mathrm{F}} = j_1\right\}, \quad i_1 \geq 1, \qquad (4)$$

$$B_n(j_1, j_2) = \Pr\left\{q_{t_k}^{\mathrm{I}} = n, q_{t_k}^{\mathrm{F}} = j_2 \mid q_{t_{k-1}}^{\mathrm{I}} = 0, q_{t_{k-1}}^{\mathrm{F}} = j_1\right\}. \qquad (5)$$

Note that $B_n(j_1, j_2)$ is the probability that the process $\{q_{t_k}^{\mathrm{I}}, q_{t_k}^{\mathrm{F}}\}_{k\geq 0}$ makes a transition from state $(0, j_1)$ to state (n, j_2) over the time interval between two successive packet service completions at Q^{I}. $A_n(j_1, j_2)$ is the probability that over the time interval between two successive service completions at Q^{I}, the phase process $\{q_{t_k}^{\mathrm{F}}\}_{k\geq 0}$ makes a transition from state j_1 to state j_2, and there are n packet arrivals to Q^{I}, given that

Q^{I} is non-empty at the beginning of the transition. Let $A_n(j_1, j_2, l)$ ($B_n(j_1, j_2, l)$) be defined similarly to $A_n(j_1, j_2)$ ($B_n(j_1, j_2)$) but under the additional requirement that the duration of a transition d_k is equal to l slots, $l \geq 1$:

$$A_n(j_1, j_2, l) = \Pr\left\{q^{\mathrm{I}}_{t_k} = i_1 - 1 + n, q^{\mathrm{F}}_{t_k} = j_2, d_k = l \mid q^{\mathrm{I}}_{t_{k-1}} = i_1, q^{\mathrm{F}}_{t_{k-1}} = j_1\right\}, \quad i_1 \geq 1. \tag{6}$$

$$B_n(j_1, j_2, l) = \Pr\left\{q^{\mathrm{I}}_{t_k} = n, q^{\mathrm{F}}_{t_k} = j_2, d_k = l \mid q^{\mathrm{I}}_{t_{k-1}} = 0, q^{\mathrm{F}}_{t_{k-1}} = j_1\right\}. \tag{7}$$

Let $p^{\mathrm{F}}_{\tau}(i, j, n) = \Pr\{q^{\mathrm{F}}_{t_d+n} = j, \tau(q^{\mathrm{F}}_{t_d}) = n \mid q^{\mathrm{F}}_{t_d} = i\}$, $i, j \in R^{\mathrm{F}}$, $0 \leq n \leq \mathcal{P}_i$, denote the transition probabilities of $\{q^{\mathrm{F}}_j\}_{j\geq 0}$ over a service horizon of length n slots. These are determined by the adopted service policy $\bar{\mathcal{P}}$ and the associated state-policies $\mathcal{P}_i$, $i \in R^{\mathrm{F}}$, and can be expressed in terms of the one-step transition probabilities of $\{q^{\mathrm{F}}_j\}_{j\geq 0}$ given by probability matrix P_s [14]. The stochastic matrix with elements $p^{\mathrm{F}}_{\tau}(i, j, n)$ is denoted by $P^{\mathrm{F}}_{\tau}(n)$. It can be easily shown that

$$A_n(j_1, j_2, l) = p^{\mathrm{I},l}_a(n) \sum_{j=0}^{N} p^{\mathrm{F}}_{\tau}(j_1, j, l-1) p_v(j, j_2), \quad l \geq 1. \tag{8}$$

$$\begin{aligned} B_n(j_1, j_2, l) = {} & \sum_{m=1}^{l-1} [p^{\mathrm{I}}_a(0)]^{m-1} \sum_{j=0}^{N} p^{\mathrm{F}}_q(j_1, j, m) \sum_{i=1}^{n+1} p^{\mathrm{I}}_a(i) p^{\mathrm{I},l-m}_a(n-i+1) \\ & \times \sum_{j'=0}^{N} p^{\mathrm{F}}_{\tau}(j, j', l-m-1) p_v(j', j_2), \quad l \geq 2. \end{aligned} \tag{9}$$

$$B_n(j_1, j_2, 1) = 0.$$

From (8) and (9) matrices $A_n(l)$ and $B_n(l)$, $l \geq 1$, may be written as

$$A_n(l) = p^{\mathrm{I},l}_a(n) P^{\mathrm{F}}_{\tau}(l-1) P_v, \tag{10}$$

$$B_n(l) = \sum_{m=1}^{l-1} [p^{\mathrm{I}}_a(0)]^{m-1} P^{\mathrm{F}}_q(m) \sum_{i=1}^{n+1} p^{\mathrm{I}}_a(i) A_{n-i+1}(l-m), \qquad B_n(1) = 0. \tag{11}$$

Application of the law of total probability yields

$$A_n = \sum_{l=1}^{\infty} A_n(l) = \sum_{l=1}^{\infty} p^{\mathrm{I},l}_a(n) P^{\mathrm{F}}_{\tau}(l-1) P_v, \tag{12}$$

$$B_n = \sum_{l=1}^{\infty} B_n(l) = [I - p^{\mathrm{I}}_a(0) P_s]^{-1} P_s \sum_{i=1}^{n+1} p^{\mathrm{I}}_a(i) A_{n-i+1}. \tag{13}$$

The stationary joint probabilities for the queue occupancies upon packet service completions at Q^{I}, $y(i, j)$, $i \in R^{\mathrm{I}}$, $j \in R^{\mathrm{F}}$, as determined by the transition probability matrix $P_{(q^{\mathrm{I}}, q^{\mathrm{F}})}$, are derived by applying matrix analytic techniques [14]. The stationary joint probabilities at arbitrary time instants, $\psi(i, j)$, $i \in R^{\mathrm{I}}$, $j \in R^{\mathrm{F}}$ are then obtained in terms of $y(i, j)$ by invoking Markov renewal theory arguments [14].

4. Partial performance ordering of policies in $\mathcal{S}$

In view of the structure of class $\mathcal{S}$ it is possible to establish a partial performance ordering of policies in $\mathcal{S}$, as shown below.

Definition 4. Consider service policies $\bar{\Sigma} = \{\Sigma_1, \ldots, \Sigma_N\}$ and $\bar{\Omega} = \{\Omega, \ldots, \Omega_N\}$ in $\mathcal{S}$. Let $q_k^{M,\bar{\mathcal{P}}}$ denote the occupancy of Q^M at time slot k under policy $\bar{\mathcal{P}}$, $M \in \{\mathrm{I}, \mathrm{F}\}$, $\bar{\mathcal{P}} \in \{\bar{\Omega}, \bar{\Sigma}\}$. Policy $\bar{\Omega}$ is said to outperform policy $\bar{\Sigma}$ with respect to Q^M if and only if $q_k^{M,\bar{\Omega}} \le q_k^{M,\bar{\Sigma}}$, $M \in \{\mathrm{I}, \mathrm{F}\}$, for all $k \ge 0$. That is,

$$\bar{\Sigma} \overset{M}{\prec} \bar{\Omega} \iff q_k^{M,\bar{\Omega}} \le q_k^{M,\bar{\Sigma}}, \quad M \in \{\mathrm{I}, \mathrm{F}\}. \tag{14}$$

Notice that (14) implies that if $\bar{\Sigma} \overset{\mathrm{F}}{\prec} \bar{\Omega}$, then

$$L^{\mathrm{F},\bar{\Omega}} \le L^{\mathrm{F},\bar{\mathcal{P}}} \quad \text{and} \quad D^{\mathrm{I},\bar{\Omega}} \ge D^{\mathrm{I},\bar{\mathcal{P}}}, \tag{15}$$

where $L^{\mathrm{F},\bar{\mathcal{P}}}$ denotes the packet loss probability at Q^{F} under policy $\bar{\mathcal{P}}$ and $D^{\mathrm{I},\bar{\mathcal{P}}}$ denotes the average packet delay at Q^{I} under policy $\bar{\mathcal{P}}$, $\bar{\mathcal{P}} \in \{\bar{\Sigma}, \bar{\Omega}\}$. The following proposition provides for a partial performance ordering of policies in $\mathcal{S}$; its proof may be found in Appendix A.

Proposition 2. *Consider two policies* $\bar{\Sigma} = \{\Sigma_1, \ldots, \Sigma_N\}$ *and* $\bar{\Omega} = \{\Omega_1, \ldots, \Omega_N\}$ *in* $\mathcal{S}$. *Then*

$$\Sigma_i \le \min_{1 \le k \le N} \{\Omega_k\} \qquad \text{for all } i,\ 1 \le i \le N, \tag{16}$$

implies that

$$\bar{\Sigma} \overset{\mathrm{F}}{\prec} \bar{\Omega} \quad \text{and} \quad \bar{\Omega} \overset{\mathrm{I}}{\prec} \bar{\Sigma}. \tag{17}$$

Corollary 1. *Let* $\bar{\mathcal{L}}_k = \{k, k, \ldots, k\}$ *denote the k-limited service policy in* $\mathcal{S}$, $0 \le k \le \infty$. *Then Proposition 2 implies that*

$$\text{if } k \le j \quad \text{then } \bar{\mathcal{L}}_k \overset{\mathrm{F}}{\prec} \bar{\mathcal{L}}_j \text{ and } \bar{\mathcal{L}}_j \overset{\mathrm{I}}{\prec} \bar{\mathcal{L}}_k. \tag{18}$$

Under some conditions on the arrival process to Q^{F} a weaker than (16) sufficient condition for performance ordering of policies in $\mathcal{S}$ may be obtained as described in Proposition 3.

Proposition 3. *Consider two policies* $\bar{\Sigma} = \{\Sigma_1, \ldots, \Sigma_N\}$ *and* $\bar{\Omega} = \{\Omega_1, \ldots, \Omega_N\}$ *in* $\mathcal{S}$. *Let* $a^{\mathrm{F}}(\Delta t)$ *denote the number of packet arrivals to* Q^{F} *over an interval of length* Δt *(slots). If*

$$a^{\mathrm{F}}(\Delta t) \le \Delta t \tag{19}$$

for all Δt, $\Delta t \in \{1, 2, \ldots\}$, *and*

$$\Sigma_i \le \Omega_k \qquad \text{for all } i \ge k \text{ and } k,\ 1 \le k \le N, \tag{20}$$

then

$$\bar{\Sigma} \overset{\mathrm{F}}{\prec} \bar{\Omega} \quad \text{and} \quad \bar{\Omega} \overset{\mathrm{I}}{\prec} \bar{\Sigma}. \tag{21}$$

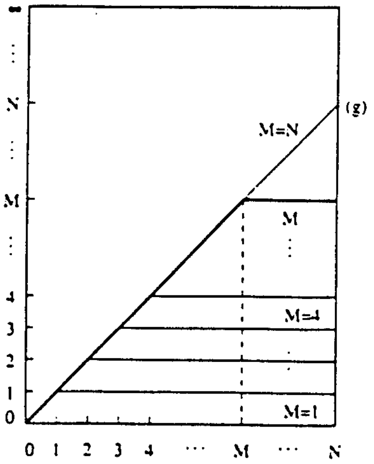


Fig. 4. Policy-line representation of the set of policies in Example 1.

The proof of Proposition 3 may be found in Appendix A.

When the arrrival process to Q^{F} is Bernoulli, condition (19) in Proposition 3 holds and thus, the following corollary is evident in view of Proposition 3.

Corollary 2. *Consider two policies* $\bar{\Sigma} = \{\Sigma_1, \ldots, \Sigma_N\}$ *and* $\bar{\Omega} = \{\Omega_1, \ldots, \Omega_N\}$ *in* $\mathcal{S}$. *Let the packet arrival process to* Q^{F} *be a Bernoulli process. If*

$$\Sigma_i \leq \Omega_k \quad \text{for all } i \geq k, \ 1 \leq k \leq N, \tag{22}$$

then

$$\bar{\Sigma} \overset{\mathrm{F}}{\prec} \bar{\Omega} \quad \text{and} \quad \bar{\Omega} \overset{\mathrm{I}}{\prec} \bar{\Sigma}. \tag{23}$$

5. Examples, numerical results and discussion

The HoL priority policy for Q^{F} (upper boundary of the state-policy space $\mathcal{SP}$) minimizes packet delay/loss for this queue. Similarly, the HoL priority policy for Q^{I} (lower boundary of the state-policy space $\mathcal{SP}$) minimizes packet delay for this queue. A (state-dependent) service policy in $\mathcal{S}$ is expected to provide greater flexibility in meeting performance requirements associated with each queue. The vector representation of a service policy in $\mathcal{S}$ allows for the description of ad hoc, customized policies as illustrated in the following examples.

Example 1. The service policy $\bar{\mathcal{P}}$ defined by the following state-policies $\mathcal{P}_i$,

$$\mathcal{P}_i = \begin{cases} i, & 1 \leq i \leq M, \\ M, & M < i \leq N, \end{cases} \qquad 1 \leq M \leq N,$$

provides unlimited gated service to Q^{F} when the content (state) of Q^{F} is below some threshold M, $1 \leq M \leq N$, and M-limited service otherwise (Fig. 4). This policy provides increased service to Q^{I} compared

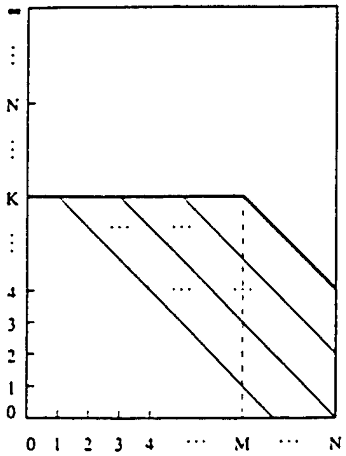


Fig. 5. Policy-line representation of the set of policies in Example 2.

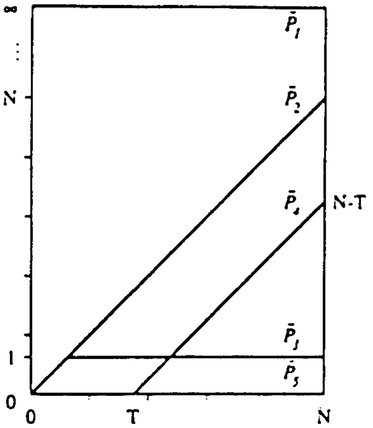


Fig. 6. Policy-line representation of the set of policies in Example 3.

to the unlimited gated policy when the content of Q^{F} exceeds the threshold M; it also provides increased service to Q^{I} compared to the M-limited policy for $1 \leq i < M$. The policy becomes 1-limited for $M = 1$ and unlimited gated for $M = N$.

Example 2. Consider the policies L_k^M which are k-limited up to a certain state M, $1 \leq M \leq N$, and then drop linearly with fixed slope, as shown in Fig. 5. If packet arrivals to Q^{F} satisfy the condition in Proposition 3, for instance, Bernoulli arrivals, then the following performance ordering holds true: $L_k^{M_1} \overset{\mathrm{F}}{\prec} L_k^{M_2}$ and $L_k^{M_2} \overset{\mathrm{I}}{\prec} L_k^{M_1}$ if $M_1 < M_2$.

Example 3. Consider the following policies (Fig. 6):

$\bar{\mathcal{P}}_1$: $\mathcal{P}_i = \infty$, $1 \leq i \leq N$ (unlimited non-gated or ∞-limited or HoL for Q^{F});
$\bar{\mathcal{P}}_2$: $\mathcal{P}_i = i$, $1 \leq i \leq N$ (unlimited gated);
$\bar{\mathcal{P}}_3$: $\mathcal{P}_i = 1$, $1 \leq i \leq N$ (1-limited);

$\bar{\mathcal{P}}_4$: $\mathcal{P}_i = 0, 1 \leq i \leq T$ and $\mathcal{P}_i = i - T, T < i \leq N$; according this service policy Q^{I} receives HoL priority service whenever the content of Q^{F} is less than a threshold T. If the content i of Q^{F} at a decision time-instant exceeds the value of the 'safety' threshold T, service is provided to $i - T$ packets at Q^{F}. Compared to the HoL priority service policy for Q^{I}, this 'mixed' policy provides increased service to the finite queue when it is critically congested ($i > T$) and is expected to decrease the loss probability.

$\bar{\mathcal{P}}_5$: $\mathcal{P}_i = 0, 1 \leq i \leq N$ (0-limited or HoL for Q^{I}).

In view of Proposition 1, the following performance ordering of the policies may be easily established:

$$\bar{\mathcal{P}}_5 \overset{\mathrm{F}}{\prec} \bar{\mathcal{P}}_3 \overset{\mathrm{F}}{\prec} \bar{\mathcal{P}}_2 \overset{\mathrm{F}}{\prec} \bar{\mathcal{P}}_1 \quad \text{and} \quad \bar{\mathcal{P}}_5 \overset{\mathrm{F}}{\prec} \bar{\mathcal{P}}_4 \overset{\mathrm{F}}{\prec} \bar{\mathcal{P}}_1.$$

$$\bar{\mathcal{P}}_1 \overset{\mathrm{I}}{\prec} \bar{\mathcal{P}}_2 \overset{\mathrm{I}}{\prec} \bar{\mathcal{P}}_3 \overset{\mathrm{I}}{\prec} \bar{\mathcal{P}}_5 \quad \text{and} \quad \bar{\mathcal{P}}_1 \overset{\mathrm{I}}{\prec} \bar{\mathcal{P}}_4 \overset{\mathrm{I}}{\prec} \bar{\mathcal{P}}_5.$$

Notice that $\bar{\mathcal{P}}_1 \overset{\mathrm{I}}{\prec} \bar{\mathcal{P}} \overset{\mathrm{I}}{\prec} \bar{\mathcal{P}}_5$ and $\bar{\mathcal{P}}_5 \overset{\mathrm{F}}{\prec} \bar{\mathcal{P}} \overset{\mathrm{F}}{\prec} \bar{\mathcal{P}}_1$ for all service policies $\bar{\mathcal{P}} \in \mathcal{S}$, which implies that $\bar{\mathcal{P}}_1$ and $\bar{\mathcal{P}}_5$ determine bounds on the achievable performance by any service policy in $\mathcal{S}$.

The performance of some service policies in $\mathcal{S}$ is evaluated in terms of the induced packet loss probability at the finite queue, L^{F}, and the average packet delays D^{I} and D^{F}. These can be easily determined from the joint probability distribution of the queue occupancies. Let λ^{F} (λ^{I}) denote the packet arrival rate to Q^{F} (Q^{I}). Then

$$L^{\mathrm{F}} = \frac{\lambda^{\mathrm{F}} + \lambda^{\mathrm{I}} - \rho'}{\lambda^{\mathrm{F}}},$$

where $\rho' = 1 - \psi(0, 0) = 1 - \Pr\{q^{\mathrm{I}} = 0, q^{\mathrm{F}} = 0\}$ is the system utilization. The average packet delays D^{I} and D^{F} are derived from the mean queue occupancies $E[q^{\mathrm{I}}]$ and $E[q^{\mathrm{F}}]$ by applying Little's theorem:

$$D^{\mathrm{I}} = \frac{E[q^{\mathrm{I}}]}{\lambda^{\mathrm{I}}}, \qquad D^{\mathrm{F}} = \frac{E[q^{\mathrm{F}}]}{\lambda^{\mathrm{F}}(1 - L^{\mathrm{F}})}.$$

Some numerical results for L^{F} and D^{I} induced by service policies $\bar{\mathcal{P}}_1 - \bar{\mathcal{P}}_5$ in Example 3 have been derived and are shown in Figs. 7 and 8 under Bernoulli and in Figs. 9 and 10 under truncated Poisson packet arrival processes to each of the queues. The finite queue capacity is $N = 29$. For $\bar{\mathcal{P}}_4$, two different threshold values are considered, $T = 10$ and $T = 15$. For each type of arrival processes the mean packet delay D^{I} and the packet loss probability L^{F} are plotted versus: (i) the arrival rate λ^{I} for a fixed arrival rate $\lambda^{\mathrm{F}} = 0.5$ (packets/slot) and a range of offered traffic load between 0.8 and 0.925 (Figs. 7 and 9) and (ii) the arrival rate λ^{F} for a fixed arrival rate $\lambda^{\mathrm{I}} = 0.5$ (packets/slot) and the same total load range as in (i) (Figs. 8 and 10).

The results shown in Figs. 7–10 are in accordance with the performance ordering established by Proposition 2. Furthermore, they illustrate the potential for improved performance provided by the ad hoc service policies in $\mathcal{S}$, such as $\bar{\mathcal{P}}_4$. For instance, if upper bounds on L^{F} equal to 10^{-12} (under Bernoulli arrivals) or 10^{-9} (under truncated Poisson arrivals) are necessary to meet quality of service requirements, then only service policies $\bar{\mathcal{P}}_1$, $\bar{\mathcal{P}}_2$ and $\bar{\mathcal{P}}_4$ ($T = 10$) are acceptable for all values of λ^{I} and λ^{F} considered in Figs. 7–10. Among these policies, $\bar{\mathcal{P}}_4$ ($T = 10$) induces significantly lower D^{I} compared to that under $\bar{\mathcal{P}}_1$ and $\bar{\mathcal{P}}_2$ and, thus, it is the most effective in the sense that it minimizes D^{I} while satisfying the constraint on L^{F}.

Results for finite capacity $N = 50$ and Poisson packet arrival processes are shown in Figs. 11 and 12. For policy $\bar{\mathcal{P}}_4$, the thresholds $T = 15$ and $T = 25$ have been considered. Results for $N = 50$ and Bernoulli

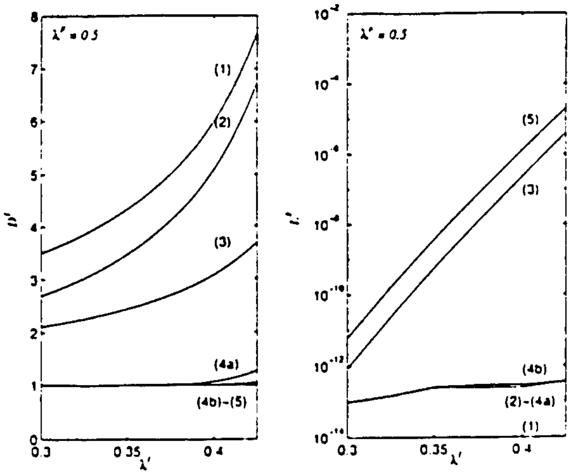


Fig. 7. Mean packet delay D^I and loss probability L^F under Bernoulli packet arrivals ($N = 29$, $\lambda^F = 0.5$). Curve (i) corresponds to policy $\bar{\mathcal{P}}_i$; (4a) and (4b) correspond to $\bar{\mathcal{P}}_4$ with $T = 10$ and $T = 15$, respectively. $(i) \sim (j)$ indicates that the results under policies $\bar{\mathcal{P}}_i$ and $\bar{\mathcal{P}}_j$ are approximately the same.

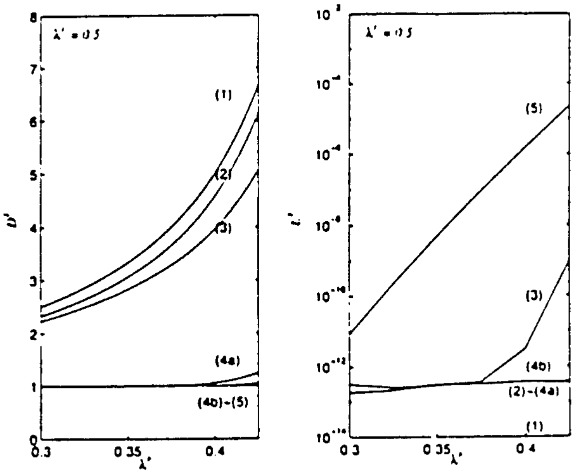


Fig. 8. Mean packet delay D^I and loss probability L^F under Bernoulli packet arrivals ($N = 29$, $\lambda^I = 0.5$). Curve (i) corresponds to policy $\bar{\mathcal{P}}_i$; (4a) and (4b) correspond to $\bar{\mathcal{P}}_4$ with $T = 10$ and $T = 15$, respectively. $(i) \sim (j)$ indicates that the results under policies $\bar{\mathcal{P}}_i$ and $\bar{\mathcal{P}}_j$ are approximately the same.

arrival processes have not been presented since most of policies $\bar{\mathcal{P}}_1$–$\bar{\mathcal{P}}_5$—with the exception of $\bar{\mathcal{P}}_3$ and $\bar{\mathcal{P}}_4$ over certain ranges of traffic loads—induce very low loss probability at Q^F. In this case, the accuracy of the derived loss probabilities depends on the numerical precision of the machine.

The queueing system introduced and studied in this work could model the queueing behavior of a protocol which allocates a common resource to two traffic classes with different Quality-of-Service (QoS) requirements. One class may be viewed as a best effort class [15] and the second as having more stringent QoS requirements. A sufficiently large buffer (Q^I) can be assumed to be available to the best effort traffic—called also Available Bit Rate (ABR)—for which a mean value of the induced queueing intensity (such

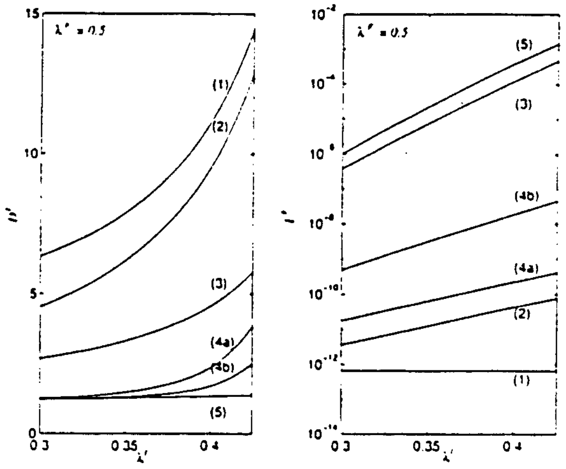


Fig. 9. Mean packet delay D^I and loss probability L^F under Poisson packet arrivals ($N = 29$, $\lambda^F = 0.5$). Curve (i) corresponds to policy $\bar{\mathcal{P}}_i$; (4a) and (4b) correspond to $\bar{\mathcal{P}}_4$ with $T = 10$ and $T = 15$, respectively.

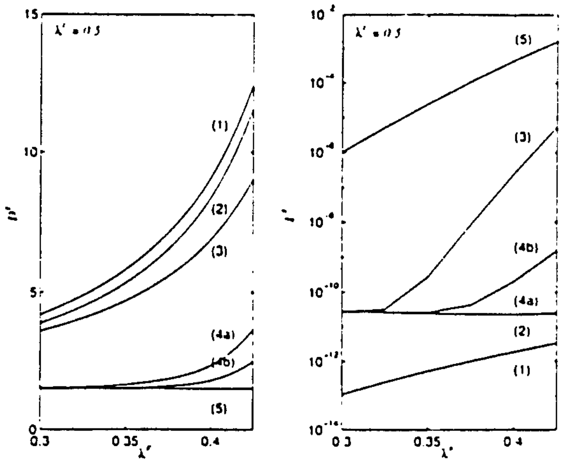


Fig. 10. Mean packet delay D^I and loss probability L^F under Poisson packet arrivals ($N = 29$, $\lambda^I = 0.5$). Curve (i) corresponds to policy $\bar{\mathcal{P}}_i$; (4a) and (4b) correspond to $\bar{\mathcal{P}}_4$ with $T = 10$ and $T = 15$, respectively.

as delay) would be meaningful. For the traffic with more stringent QoS requirements a finite buffer (Q^F) could be considered to capture, for instance, a requirement associated with a maximum tolerable delay. For example, if a packet delay exceeding N (slots) is unacceptable, then it is evident that a buffer capacity larger than N is meaningless. Furthermore, the cell loss probability would serve as a lower bound on the probability that a packet is delayed beyond N slots. Under the HoL priority policy for Q^F, the packet loss probability and the probability that a packet is delayed by more than N slots will coincide, assuming that such packets do not receive service.

Finally, the partial ordering of the proposed policies can provide for an efficient search for a policy which delivers a desired performance. At first, the Head-of-Line (HoL) priority for one class would establish if there exists *any* policy in the class which would induce the target-performance (or desired Quality-of-

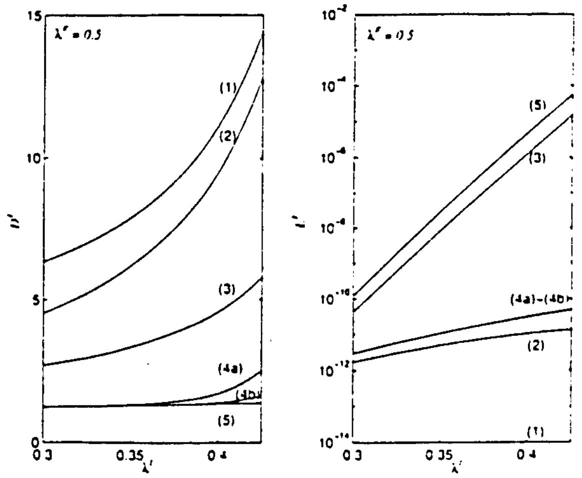


Fig. 11. Mean packet delay D^I and loss probability L^F under Poisson packet arrivals ($N = 50$, $\lambda^F = 0.5$). Curve (i) corresponds to policy $\bar{\mathcal{P}}_i$; (4a) and (4b) correspond to $\bar{\mathcal{P}}_4$ with $T = 15$ and $T = 25$, respectively.

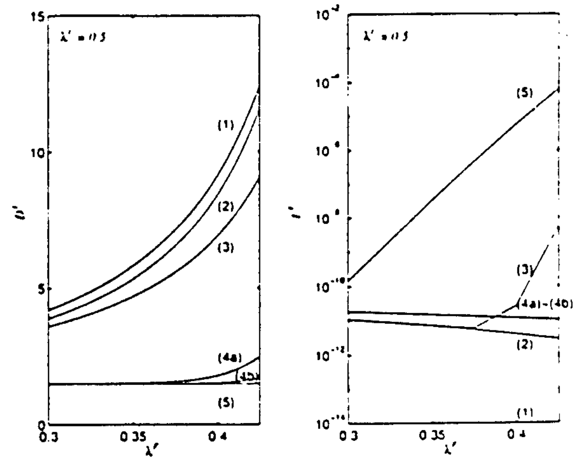


Fig. 12. Mean packet delay D^I and loss probability L^F under Poisson packet arrivals ($N = 50$, $\lambda^I = 0.5$). Curve (i) corresponds to policy $\bar{\mathcal{P}}_i$; (4a) and (4b) correspond to $\bar{\mathcal{P}}_4$ with $T = 15$ and $T = 25$, respectively.

Service) for this class. If it exists, then Proposition 2 could provide guidance in the search for an acceptable policy. It may not identify the best policy or even find an existing acceptable policy, but it is one—possibly effective—alternative to the exhaustive search. By considering different starting policies, the entire space could be searched in a non-exhaustive manner by moving 'toward' the better policies as indicated by Proposition 2.

6. Conclusions

The main contribution of this paper is the introduction and study of the versatile class of service policies $\mathcal{S}$ based on the concept of the state-policy. Through the construction of service policies in terms of policy-lines on the space of state-policies $\mathcal{SP}$ (Fig. 2), the flexibility in designing policies in $\mathcal{S}$ is illustrated. Through the introduced simple performance ordering of service policies a direction for the design of service policies in $\mathcal{S}$ whose performance approach a desired level is presented. This indicates the policy design flexibility and potential for closely achieving a desired level of performance associated with the introduced class of service policies. The latter have been illustrated through some examples and numerical results.

Appendix A. Proof of Propositions 2 and 3

A.1. *Proof of Proposition 2*

It suffices to show that under any realization of the packet arrival process

$$q_k^{\mathrm{F},\bar{\Omega}} \le q_k^{\mathrm{F},\bar{\Sigma}} \quad (\text{or } q_k^{\mathrm{I},\bar{\Sigma}} \le q_k^{\mathrm{I},\bar{\Omega}})$$

for all $k \ge 0$. Consider an initially empty system. Let t_1 mark the (end of the time) slot at which the first packet arrives to Q^{I}; clearly,

$$q_{t_1}^{\mathrm{F},\bar{\Omega}} = q_{t_1}^{\mathrm{F},\bar{\Sigma}},$$

since uninterrupted service to Q^{F} has been provided under both policies up to t_1. Condition (16) implies that the service horizons under policies $\bar{\Sigma}$ and $\bar{\Omega}$—denoted by $\tau^{\bar{\Sigma}}(q_{t_1}^{\mathrm{F},\bar{\Sigma}})$ and $\tau^{\bar{\Omega}}(q_{t_1}^{\mathrm{F},\bar{\Omega}})$, respectively—will satisfy

$$\tau^{\bar{\Sigma}}(q_{t_1}^{\mathrm{F},\bar{\Sigma}}) \le \tau^{\bar{\Omega}}(q_{t_1}^{\mathrm{F},\bar{\Omega}}),$$

and thus,

$$q_t^{\mathrm{I},\bar{\Sigma}} \le q_t^{\mathrm{I},\bar{\Omega}}, \quad t_1 \le t \le t_2^{\bar{\Omega}},$$

since a packet from Q^{I} is served under policy $\bar{\Sigma}$ no latter than under policy $\bar{\Omega}$; $t_2^{\bar{\Omega}} = t_1 + \tau^{\bar{\Omega}}(q_{t_1}^{\mathrm{F},\bar{\Omega}}) + 1$ marks the slot at which the server serves the first packet in Q^{I} under policy $\bar{\Omega}$. At $t_2^{\bar{\Omega}}$ the following cases may be distinguished.

(a) If $q_{t_2^{\bar{\Omega}}}^{\mathrm{I},\bar{\Omega}} = 0$, then it is easy to see that $q_{t_2^{\bar{\Omega}}}^{\mathrm{I},\bar{\Sigma}} = 0$, since the former implies that only one packet arrival to Q^{I} has occurred up to $t_2^{\bar{\Omega}}$. Let t_1' mark the arrival time of the next packet to Q^{I}. Then,

$$q_t^{\mathrm{I},\bar{\Sigma}} = q_t^{\mathrm{I},\bar{\Omega}} = 0, \quad t_2^{\bar{\Omega}} \le t < t_1'.$$

Thus, it has been proven that

$$q_t^{\mathrm{I},\bar{\Sigma}} \le q_t^{\mathrm{I},\bar{\Omega}}, \quad t_1 \le t < t_1',$$

where $[t_1, t_1')$ denotes the time interval of an I-cycle, defined to be the interval between consecutive arrivals to Q^{I} which find the buffer empty under both policies $\bar{\Sigma}$ and $\bar{\Omega}$.

(b) If $q^{\mathrm{I}.\hat{\Omega}}_{t_2^{\hat{\Omega}}} > 0$, then it is easy to see that the time t_0 at which the second packet is forwarded to the head of Q^{I} under policy $\bar{\Sigma}$ will satisfy

$$t_1 + \tau^{\bar{\Sigma}}(q^{\mathrm{F}.\bar{\Sigma}}_{t_1}) + 1 \triangleq t_2^{\bar{\Sigma}} \le t_0 \le t_2^{\hat{\Omega}}.$$

The departure time instant of the second packet under policies $\bar{\Sigma}$ and $\hat{\Omega}$, will be

$$t_3^{\bar{\Sigma}} \triangleq t_0 + \tau^{\bar{\Sigma}}(q^{\mathrm{F}.\bar{\Sigma}}_{t_0}) + 1 \quad \text{and} \quad t_3^{\hat{\Omega}} \triangleq t_2^{\hat{\Omega}} + \tau^{\hat{\Omega}}(q^{\mathrm{F}.\hat{\Omega}}_{t_2^{\hat{\Omega}}}) + 1,$$

respectively. Since in view of (16),

$$\tau^{\bar{\Sigma}}(q^{\mathrm{F}.\bar{\Sigma}}_{t_0}) \le \tau^{\hat{\Omega}}(q^{\mathrm{F}.\hat{\Omega}}_{t_2^{\hat{\Omega}}}), \tag{A.1}$$

it is implied that $t_3^{\bar{\Sigma}} \le t_3^{\hat{\Omega}}$ and

$$q_t^{\mathrm{I}.\bar{\Sigma}} \le q_t^{\mathrm{I}.\hat{\Omega}}, \quad t_0 \le t \le t_3^{\hat{\Omega}}. \tag{A.2}$$

By reiterating the argument presented above until Q^{I} becomes empty under both policies, it is established that (A.2) holds at any time instant of an I-cycle with an arbitrary number of packet arrivals to Q^{I}.

Parts (a) and (b) above complete the proof of Proposition 2.

A.2. *Proof of Proposition 3*

The proof can be established by following the proof of Proposition 2. The introductory part and part (a) of that proof is directly applicable. In order for part (b) to be valid, the validity of (A.1) needs to be established. While condition (16) in Proposition 2 guarantees that for any values of $q^{\mathrm{F}.\bar{\Sigma}}_{t_0}$ and $q^{\mathrm{F}.\hat{\Omega}}_{t_2^{\hat{\Omega}}}$ inequality (A.1) will hold true, condition (20) in Proposition 3 would also guarantee it provided that

$$q^{\mathrm{F}.\bar{\Sigma}}_{t_0} \ge q^{\mathrm{F}.\hat{\Omega}}_{t_2^{\hat{\Omega}}}. \tag{A.3}$$

That is, provided that the finite queue occupancy when the second packet is forwarded to the head of Q^{I} under policy $\bar{\Sigma}$ be not less than under policy $\hat{\Omega}$. Since Q^{F} will have been reduced by $t_2^{\hat{\Omega}} - t_0$ at $t_2^{\hat{\Omega}}$ under policy $\hat{\Omega}$ compared to its value at t_0 under policy $\bar{\Sigma}$ and the potential new packet arrivals to Q^{F} over the interval $(t_0, t_2^{\hat{\Omega}})$ will be at most $t_2^{\hat{\Omega}} - t_0$ (condition (19)), it is evident that (A.3) will hold true. Reiterating the argument for the packet arrival to the head of Q^{I} that may follow, the proof of Proposition 3 can be established.

References

[1] H. Takagi, Queueing analysis of polling models, *ACM Comput. Surveys* 20(1) (1988) 5–28.

[2] H. Levy and M. Sidi, Polling systems: Applications, modelling, and optimization, *IEEE Trans. Comm.* 38(10) (1990) 1750–1760.

[3] H. Takagi, *Queueing analysis: A foundation of performance evaluation. Vacation and priority systems*, Vol. 1, Part 1, 42(2) (1994) 615–624.

[4] I. Stavrakakis, Delay bounds on a queueing system with consistent priorities, *IEEE Trans. Comm.*, 42(2) (1994) 615–624. *Proc. IEEE INFOCOM'92*, Florence, Italy, May 4–8 (1992) pp. 2151–2160.

[5] C. Bisdikian, A queueing model for a data station within the IEEE 802.6 MAN, *IEEE 17th Local Computer Networks Conf.*, Minneapolis, MN, September 13–16, 1992.

[6] R. Landry and I. Stavrakakis, A three-priority queueing policy with applications to communication networks, *Proc. INFOCOM'93*, San Francisco, CA, March 30–April 1, 1993, pp. 1067–1074.

[7] M. Eisenberg, Two queues with alternating service, *SIAM J. Appl. Math.* 36(2) (1979) 287–303.

[8] T. Ozawa, Alternating service queues with mixed exhaustive and K-limited services, *Perform. Eval.* 11 (1990) 165–175.

[9] I. Stavrakakis and S. Tsakiridou, Occupancy distribution for a DQDB station based on a queueing system with Markov-structured service requirements, *Proc. INFOCOM'93*, San Francisco, CA, March 30–April 1, 1993, pp. 1083–1090.

[10] C.M. Harris and W.G. Marchal, State dependence in M/G/1 server-vacation models, *Oper. Res.* 36(4) (1988) 560–565.

[11] J. Keilson and L.D. Servi, Blocking probability for M/G/1 vacation systems with occupancy level dependent schedules, *Oper. Res.* 37(1) (1989) 134–140.

[12] D.M. Lucantoni, K.S. Meier-Hellstern and M.F. Neuts, A single-server queue with server vacations and a class of non-renewal arrival processes, *Adv. in Appl. Probab.* 22 (1990) 676–705.

[13] K.K. Leung, Cyclic-service systems with probabilistically-limited service, *IEEE J. Select. Areas Commun.* 9(2) (1991) 185–193.

[14] S. Tsakiridou and I. Stavrakakis, Study of a class of partially ordered service strategies for a system of two discrete-time queues, Technical Report, TR-CDSP-96-37, Communications and Digital Signal Processing (CDSP) Center, ECE Department, Northeastern University, Boston, 1996.

[15] D. Towsley, Providing quality of service in packet switched networks, in: L. Donatiello and R. Nelson (Eds.) *Performance Evaluation of Computer and Communication Systems*, Lecture Notes in Computer Science, Vol. 729, Springer, Berlin (1993) pp. 560–586.

[16] M.F. Neuts, *Structured Stochastic Matrices of M/G/1 Type and their Applications*, Marcel Dekker, New York (1989).

Ioannis Stavrakakis received the Diploma in Electrical Engineering from the Aristotelian University of Thessaloniki, Thessaloniki, Greece, 1983, and the Ph.D. degree in Electrical Engineering from the University of Virginia, 1988.

In 1988, he joined the faculty of Computer Science and Electrical Engineering of the University of Vermont as an assistant and then associate professor. Since 1994, he has been an associate professor of Electrical and Computer Engineering of the Northeastern University, Boston. His research interests are in stochastic system modeling, teletraffic analysis and discrete-time queueing theory, with primary focus on the design and performance evaluation of Broadband Integrated Services Digital Networks (B-ISDN).

Dr. Stavrakakis is a senior member of IEEE and a member of the IEEE Communications Society, Technical Committee on Computer Communications. He has organized and chaired sessions, and has been a technical committee member, for conferences such as GLOBECOM, ICC and INFOCOM.

Sophia Tsakiridou received the Diploma in Electrical Engineering from the Aristotelian University of Thessaloniki, Greece, in 1988 and the Ph.D. degree in Electrical Engineering from the University of Vermont in 1994.

She is currently a Postdoctoral Fellow at Queen's University, Canada. Her research interests are in the areas of stochastic modeling and performance evaluation of communication networks.

Reprinted from

COMPUTER NETWORKS and ISDN SYSTEMS

Computer Networks and ISDN Systems 29 (1997) 293-313

Multiplexing ATM traffic streams with time-scale-dependent arrival processes [1]

Randall Landry [a,2], Ioannis Stavrakakis [b,*]

[a] Core Network Technology, DSP R & D Center, Texas Instruments Inc., Dallas, TX 75265, USA

[b] Department of Electrical and Computer Engineering, 409 Dana Research Building, Northeastern University Boston, MA 02115, USA

Accepted 30 June 1996

ELSEVIER

ELSEVIER

Computer Networks and ISDN Systems 29 (1997) 293–313

COMPUTER NETWORKS and ISDN SYSTEMS

Multiplexing ATM traffic streams with time-scale-dependent arrival processes [1]

Randall Landry [a,2], Ioannis Stavrakakis [b,*]

[a] *Core Network Technology, DSP R & D Center, Texas Instruments Inc., Dallas, TX 75265, USA*

[b] *Department of Electrical and Computer Engineering, 409 Dana Research Building, Northeastern University Boston, MA 02115, USA*

Accepted 30 June 1996

Abstract

This paper presents a flexible traffic model which is capable of describing traffic dynamics on a variety of time-scales associated with broadband packet networks such as ATM. The model is applied to the study of Non-Deterministic Periodic (NDP) sources as well as Variable Bit Rate (VBR) video sources in which cells are delivered at a sub-frame level referred to as a *slice*. An efficient numerical technique is presented for the study of a finite-capacity multiplexer under an input process determined by the superposition of N of the proposed traffic models. Numerical results support the assertion that the time-scales at which variations in a cell arrival stream occur have a significant impact on multiplexer performance, thereby illustrating the relevance of the proposed modeling and analytical techniques.

Keywords: Time-scales; Traffic modeling; Queueing analysis; Periodic sources; ATM

1. Introduction

The desire to accommodate a diverse mixture of user traffic in a Broadband Integrated Services Digital Network (B-ISDN) has led to an intense concentration of research efforts on technologies which allow for traffic multiplexing, such as the Asynchronous Transfer Mode (ATM). Performance evaluation of statistical multiplexers under a variety of input processes and/or service policies (see [1–6] and the references therein) has been essential in determining the achievable effectiveness of statistical multiplexing.

The development of efficient congestion control and call admission procedures is an area in which multiplexing analysis and network traffic characterization continue to play an important role. Consequently, the formulation of accurate traffic models which lend themselves to analytical studies is a topic of ongoing research. In a high-speed integrated services networking environment, accurate traffic models should have the ability to model variations in the input cell arrival process on a variety of different time-scales. These time-scales may represent time-intervals (in network time slots) over which logical information blocks are delivered to the network.

[*] Corresponding author. Email: ioannis@cdsp.neu.edu.

[1] Research supported by the Advanced Research Project Agency under Grant F49620-93-1-0564 monitored by the Air Force Office of Scientific Research (AFOSR). I. Stavrakakis is on leave from the University of Vermont.

[2] E-mail: landry@hc.ti.com.

PII S0169-7552(96)00097-9

For instance, consider a Variable Bit Rate (VBR) video coder [7–9]. Bits generated over a frame form a logical block of information which could provide a basis for the description of the generated traffic as well as the desirable QoS. Notice that if the VBR video coder pre-buffers and uniformly distributes cells in each frame, then the cell activity over a frame can be described in terms of a fixed cell-arrival rate [9], and a relevant time-scale may be that of a video frame. If pre-buffering at the frame level is not feasible (or desirable), due to the induced frame delay, pre-buffering and uniform spreading over a shorter time-scale (referred to as a slice) may be necessary. In this case, the cell arrival rate over a slice will be constant and changes in the cell arrival rate process will occur within the frame at slice boundaries. Note that the absence of frame-level pre-buffering has been shown [10–13] to give rise to cell arrival autocorrelation functions which are *pseudo-periodic*, as opposed to the monotonic autocorrelation functions arising from frame-level pre-buffering schemes [7]. Packetized voice provides yet another example of a common traffic source time-scale when one considers the duration of time required to generate enough bits to form a cell. The resulting time-scale is then representative of the minimum cell inter-arrival time associated with the source.

When traffic sources such as those mentioned above are multiplexed onto a network link of capacity C, the time-scale T associated with each source can be seen as the one at which the network perceives events of interest (i.e. frame, slice, cell arrivals) to occur, and can be easily expressed in units of slots. For instance, when a voice source generating information at a peak rate of r bits/sec is multiplexed onto an ATM link with capacity C bits/sec, a relevant time-scale for the source is given by the ratio C/r.

In this paper, a stochastic traffic model is presented which is capable of capturing the cell arrival dynamics associated with a wide range of deterministic time-scales. The model is based on an underlying m-state periodic Markov chain in which state transitions occur every $T_s \geqslant 1$ time slots. While in a given state, cells are generated independently from slot to slot according to an arbitrary distribution. Two time-scales of length T_s and mT_s may be easily identified in this traffic model. A computationally tractable performance analysis of a finite-capacity multiplexer is developed under input traffic determined by the superposition of N such traffic sources. The technique involves the decomposition of a potentially large irreducible Markov chain into smaller irreducible Markov chains whose dimensionality is independent of m and T_s. Since the time-scales T_s and mT_s are expected to increase as the network speed increases, this decomposition is central to the tractability of the analysis approach in a high-speed networking environment. The model is applied to the study of Non-Deterministic Periodic (NDP) cell streams as well as to VBR video which is multiplexed at slice level.

In the following section, the proposed traffic model is described in detail in terms of a generalized periodic Markov chain and the finite capacity multiplexer is analyzed in discrete time. Subsequent sections will present the applications mentioned above, as well as a discussion of relevant past work.

2. Source characterization and queueing analysis

In the next subsection the proposed traffic model is described in terms of a generalized periodic Markovian source and the generated traffic is characterized. The resulting sub-period-level autocorrelation function for a source is derived in Section 2.2. In Section 2.3, the superposition of N such traffic sources is considered, and the aggregate traffic is described. Finally, a finite-capacity multiplexer fed by N generalized periodic Markovian sources is studied in the last subsection. The fundamental time unit is taken to be the transmission time of the fixed size cell, or the duration of a time slot on the output link as denoted by Δ (in seconds).

2.1. Description of the proposed traffic model

Let $\{m^k, T_s^k, f_j^k(\cdot) \mid 1 \leqslant j \leqslant m^k\}$ denote a set of parameters associated with the kth traffic source, $1 \leqslant k \leqslant N$, where N denotes the total number of traffic sources to be multiplexed and the parameters are defined as follows:

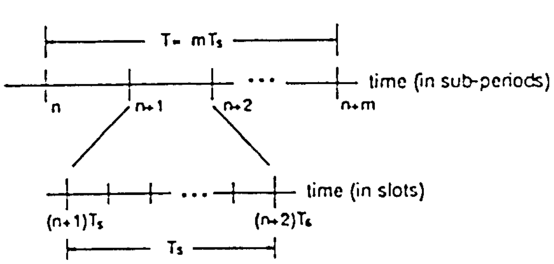


Fig. 1. The relevant time-scales associated with the GPM source.

m^k: The number of states in an underlying periodic Markov chain associated with the kth source. Let π_i^k denote the stationary probability that this periodic chain is in state i, and $p_{i,j}^{(n)}$ denote the n-step transition probability from state i to state j.

T_s^k: An integer constant (called sub-period) which represents the sojourn time (in slots) for each state in the periodic Markov chain. In other words, a transition occurs every T_s^k time slots (or $T_s^k\Delta$ seconds), and the kth source has a period of length $T^k = m^k T_s^k$ slots.

$f_j^k(\cdot)$: The probability mass function (PMF) for the number of cells generated in each slot by the kth source when that source is in state j, $1 \leqslant j \leqslant m^k$. λ_i^kj will denote the mean number of cell arrivals per slot for source k in state j, and N_j^k will denote the maximum number of such arrivals. N_j^k takes values from $\mathbb{Z}^0$, where $\mathbb{Z}^0$ denotes the set of non-negative integers.

A traffic source described in terms of $\{m^k, T_s^k, f_j^k(\cdot) | 1 \leqslant j \leqslant m^k\}$ will be referred to here as a Generalized Periodic Markov (GPM) source. It may easily be established that the mean arrival rate for source k is given by

$$\lambda^k = \frac{1}{m^k} \sum_{j=1}^{m^k} \lambda_j^k. \tag{1}$$

For the remainder of the paper, it will be assumed that all sources have the same sub-periods; hence, T_s^k will be denoted simply by T_s. In addition to the parameters defined above, let c^k (referred to as the initialization time) denote the time instant, defined at the beginning of a sub-period, at which the kth source was activated (from state 1), assuming that $1 \leqslant c^k \leqslant m^k$.

Let A_n^k denote the number of cells arrived over sub-period n, $n \geqslant 1$ (interval $[(n-1)T_s, nT_s]$); γ_n^k will denote the mean arrival rate over this interval. Arrivals are assumed to be declared at the end of the time slot over which they occur and the mean arrival rate over the nth sub-period is given by

$$\gamma_n^k \triangleq \mathrm{E}\{A_n^k\} = \lambda_j^k \quad \text{where } j = n \bmod (m^k + 1) - c^k + 1, \tag{2}$$

and n is assumed to be greater than or equal to c^k. Clearly, if $n < c^k$, then the source has not yet been activated, and $\gamma_n^k = 0$.

Note that the proposed source is capable of modeling the cell arrival dynamics of integrated services traffic sources at time-scales of one, T_s^k (sub-period) and T^k (period) slots (see Fig. 1). Since T_s can be greater than or equal to 1, the proposed traffic model is capable of capturing variations in the input process on many different time-scales. When $T_s = 1$, for instance, the proposed source model reduces to the periodic Markov chain model considered in [14], where the first and second moments of the queue length process are determined for an infinite-capacity queue in discrete time.

2.2. The sub-period-level autocorrelation function

In this section, the cell arrival autocorrelation function for a GPM source will be considered. In order to simplify notation, the source index k will be dropped for all quantities associated with the GPM model. The normalized autocorrelation function $R(n)$, sometimes referred to as the correlation coefficient at lag n, where n is indexed at sub-period boundaries, is defined as

$$R(n) \triangleq \frac{\mathrm{E}\{A_{n'}A_{n'+n}\} - \mathrm{E}^2\{A_{n'}\}}{\mathrm{E}\{A_{n'}^2\} - \mathrm{E}^2\{A_{n'}\}}. \tag{3}$$

This function is derived in Appendix A for the case of Bernoulli arrival processes within each sub-period, and is given by the expression

$$R(n) = \frac{\dfrac{1}{m}\displaystyle\sum_{i=1}^{m}\lambda_i\lambda_j + \dfrac{1}{T_s}\left(\lambda - \dfrac{1}{m}\displaystyle\sum_{i=1}^{m}\lambda_i^2\right)\delta_n - \lambda^2}{\dfrac{1}{m}\displaystyle\sum_{i=1}^{m}\lambda_i^2 + \dfrac{1}{T_s}\left(\lambda - \dfrac{1}{m}\displaystyle\sum_{i=1}^{m}\lambda_i^2\right) - \lambda^2}, \tag{4}$$

where $j = (i + n - 1)\bmod(m) + 1$, and δ_n is the Kronecker delta.

The autocorrelation function for cells/sub-period will not be used to characterize the GPM sources until Section 3.2, where the arrival PMF $f_j(\cdot)$ for each sub-period is assumed to be Bernoulli. For arrival processes other than Bernoulli, the expression for ξ^2 given in (25) will be different and Eq. (4) will require slight modifications.

2.3. The superposition of N GPM traffic sources

Before considering a statistical multiplexer fed by a number of the proposed GPM sources, the aggregate input traffic, which is given by the superposition of the individual sources, will be characterized. A useful property of the GPM traffic model is that it is closed under superposition. That is, the superposition of N GPM sources, of equal sub-periods T_s, is also a GPM source with an underlying periodic Markov chain of some period M. Note that since the initialization time c^k is defined at sub-period boundaries, it is implicitly assumed that the sub-period boundaries are synchronized among all N sources. The sources are said to be completely synchronized when $c^1 = \cdots = c^N$.

It is easy to show that the period, M, of the GPM source associated with the aggregate source is given by

$$M = \mathrm{LCM}\{m^1, m^2, \ldots, m^N\}, \tag{5}$$

where LCM $\{\cdot\}$ denotes the least common multiple of its argument.

The PMF $\bar{f}_j(\cdot)$ associated with state j of the aggregate GPM source, $1 \leqslant j \leqslant M$, is easily evaluated through the N-fold convolution

$$\bar{f}_j = f^1_{\alpha^1} * f^2_{\alpha^2} * \cdots * f^N_{\alpha^N}. \tag{6}$$

The quantities α^k are functions of both j and the initialization constant c^k, and are given by

$$\alpha^k = \begin{cases} j + c^k - 1 & \text{if } 1 \leqslant j < c^k, \\ j - c^k + 1 & \text{if } c^k \leqslant j < c^k + m^k, \\ j - m^k - c^k + 1 & \text{if } c^k + m^k \leqslant j \leqslant M. \end{cases} \tag{7}$$

2.4. Multiplexer performance analysis

In this section a First-Come First-Served (FCFS) multiplexer fed by N GPM sources (Fig. 2) is analyzed in discrete time; its buffer capacity is assumed to be finite and is denoted by B.

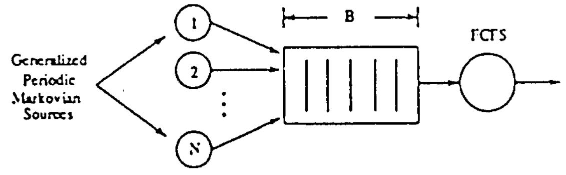


Fig. 2. The finite-capacity queueing system.

It is assumed that cell departures occur before new arrivals are counted at the end of each time slot. The system state is defined at sub-period boundaries in terms of the quantities (q_k, x_k), where q_k represents the queue length (multiplexer occupancy) at the end of sub-period k (time instant kT_s), and x_k denotes the state of the aggregate source at that time. It is easy to establish that the stochastic process $\{q_k, x_k\}_{k>1}$ is a Markov chain with state space $\{(i, j)|0 \leqslant i \leqslant B, 1 \leqslant j \leqslant M\}$, where i (j) denotes the current state of q_k (x_k). The probability that the Markov chain $\{q_k, x_k\}_{k>1}$ moves from state (i, j) at time k (defined at sub-period boundaries) to state (i', j') at time $k+1$ will be denoted by $p(i, j, i', j')$. The corresponding transition matrix is sparse due to the simple periodic structure of the input process $\{x_k\}_{k>1}$ and, in fact, a fraction of $(M-1)/M$ of the total number of elements are zero.

By mapping $p(i, j, i', j')$, for $0 \leqslant i, i' \leqslant B$ and $1 \leqslant j, j' \leqslant M$, into $p(l, l')$ through the transformation, $l = (j-1)(B+1)+i$ and $l' = (j'-1)(B+1)+i'$, the resulting transition matrix, $\mathbf{P}$, with elements $p(l, l')$, takes the canonical form of an irreducible periodic Markov chain of period M [15], given by

$$\mathbf{P} = \begin{bmatrix} 0 & \mathbf{P}_1 & 0 & \cdots & 0 \\ 0 & 0 & \mathbf{P}_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \mathbf{P}_{M-1} \\ \mathbf{P}_M & 0 & 0 & \cdots & 0 \end{bmatrix} \tag{8}$$

The following proposition allows for the easy determination of the square matrices $\mathbf{P}_j$, $1 \leqslant j \leqslant M$, which are of order $(B+1)$.

Proposition 1. *Given some initial value for q_k at the beginning of the $(k+1)$st sub-period (system state is $(q_k = i, x_k = j)$), the finite queue will evolve exactly as an* Geo/D/1/B *queue, with i.i.d. batch arrivals given by $\tilde{f}_j$, over the $(k+1)$st sub-period (interval $[kT_s, kT_s + T_s]$).* □

In view of Proposition 1, it is easy to see that the matrices $\mathbf{P}_j$ are simply the T_s-step transition probability matrices for a finite queue, under the i.i.d. batch arrival process specified by $\tilde{f}_j(\cdot)$, where $1 \leqslant j \leqslant M$. The one-step transition probabilities $p_j(i, i')$, $0 \leqslant i, i' \leqslant B$, are given below.

For $0 \leqslant i \leqslant B$ and $\max(0, i-1) \leqslant i' \leqslant B-1$,

$$p_j(i, i') = \tilde{f}_j(i'-i+1) + \tilde{f}_j(0) \cdot 1_{\{i+i'=0\}}, \tag{9}$$

where $1_{\{\cdot\}}$ denotes the indicator function.

For $0 \leqslant i \leqslant B$ and $i' = B$,

$$p_j(i, i') = \sum_{l=B-i+1}^{N_j} \tilde{f}_j(l), \tag{10}$$

where N_j denotes the maximum number of arrivals, as determined by (6), when the source is in state j. The elements of $\mathbf{P}_j$, denoted $p_j^{(T_s)}(i, i')$, can now be calculated by using the recursion

$$p_j^{(T_s)}(i, i') = \sum_l p_j^{(T_s-1)}(i, l) \cdot p_j(l, i'). \tag{11}$$

It is important to note that the dimensionality of the transition matrix $\mathbf{P}$, which is $M(B+1) \times M(B+1)$, prohibits the solution of systems with input processes having large periods, and consequently, large values of M. The following theorem, however, provides for the exact stationary solution of $\{q_k, x_k\}_{k \geq 1}$ by solving M Markov chains with transition matrices of order $(B+1)$ and thus, significantly reduces the numerical complexity involved in calculating the stationary probability vector of the transition matrix in (8). Its proof can be found in [16].

Theorem 2. *The finite irreducible periodic Markov chain, with transition matrix given by* (8), *has a unique stationary probability vector,* $\pi = (\pi_{0,1} \cdots \pi_{B,M})'$, *given by*

$$\pi = \left(\frac{1}{M}\right) \begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_M \end{bmatrix}, \tag{12}$$

where π_i *is the unique stationary probability vector for* $\mathbf{A}_i = \mathbf{P}_i \cdots \mathbf{P}_M \cdot \mathbf{P}_1 \cdots \mathbf{P}_{i-1}$ $(i = 2, \dots, M)$, *and* $\mathbf{A}_1 = \mathbf{P}_1 \cdots \mathbf{P}_M$. □

Upon solving for $\pi = (\pi_{0,1} \cdots \pi_{B,M})'$ using Theorem 2, the queue length probability distribution at sub-period boundaries is easily evaluated as

$$q(k) = \lim_{n \to \infty} \Pr\{q_n = k\} = \sum_{j=1}^{M} \pi_{k,j}, \quad k = 0, \dots, B. \tag{13}$$

Let $\hat{\pi}_{k,j}$ denote the stationary probability that the system is in state (k, j) at an arbitrary time instant (slot boundary). These quantities are derived in Appendix B through the use of a renewal/regenerative theory-based approach. The queue length distribution $\hat{q}(k)$ at arbitrary time instants can now be computed as in (13). Note that when $T_s = 1$ (sub-period is equivalent to the slot), the queue length distribution at arbitrary times is simply $\hat{q}(k) = q(k)$ for $0 \leqslant k \leqslant B$.

The cell-loss probability L is also derived in Appendix B using the same approach followed for the derivation of $\hat{\pi}_{k,j}$, and is given by

$$\begin{aligned} L &\triangleq \Pr\{\text{arriving cell is dropped}\} \\ &= \frac{1}{\lambda T_s} \sum_{i'=0}^{B} \sum_{j=1}^{M} \sum_{n=1}^{T_s} \sum_{i=I_1}^{B} \sum_{k=K_1}^{K_2} k p_j^{(n-1)}(i', i) \bar{f}_j(B+1-i+k) \pi_{i',j}, \end{aligned} \tag{14}$$

where $I_1 = \max(0, B+2-N)$, $K_1 = \max(0, i'-B-1)$, $K_2 = N-B-1+i'$ and λ denotes the total offered load, which is given by

$$\lambda = \sum_{k=1}^{N} \lambda^k. \tag{15}$$

3. Application 1: Non-Deterministic Periodic (NDP) sources

An important characteristic of high-speed packet networks such as ATM, which is not considered in the relevant performance studies of [1–6] is the potentially large disparity in the relative speeds of an output link and an input source. This disparity, or *speed-up factor* – commonly appearing at the network edges where a number of sources are being multiplexed onto an output link – results in successive cells from an active source being spaced by the interval of time required to generate enough bits to form a cell. The speed-up factor $m^k = C/r^k$, where C is the link speed (in slots/sec) and r^k is the source speed is also relevant in inter-networking, where a number of slower packet networks of various speeds are connected to the B-ISDN through a common gateway. In addition, it has been shown in [17] that, in a networking environment where input and output lines are of the same speed, source periodicities like those mentioned above give rise to cell arrival streams within the network which also exhibit significant periodicities. This observation has been used in [18] to justify the adoption of periodic input cell arrival streams for the simulation study of an ATM switch.

One way to study the speed-up factor analytically is by considering the multiplexing of deterministic periodic sources [19,20], which implies that all of the input sources are Constant Bit Rate (CBR). A more realistic traffic model for high-speed packet networks such as ATM would be a Non-Deterministic Periodic (NDP) model in which cells are generated every m^k slots according to a stochastic process, thereby allowing for a potential statistical multiplexing gain.

A relevant question that arises when considering the multiplexing of NDP sources is whether or not the aggregation of such sources can be approximated by any of the commonly used traffic models. For instance, it is widely accepted that the superposition of a large number of uncorrelated cell arrival streams can be approximated by a Poisson process, which allows for a simple analytical solution to many queueing problems. However, this limiting assumption may not be valid when the cell arrival processes have a periodic structure. In fact, it will be shown that for the superposition of NDP sources, the Poisson assumption can be either pessimistic or optimistic, depending upon the operating conditions.

A correlated traffic model in which cells are generated at a minimum spacing of s^k slots is considered in [21] and [22], where the cell arrival process is governed by an underlying 2-state Markov chain. The speed-up factor s^k, which is equal to the unit step transition time of the Markov chain for source k, is shown to have a significant impact on multiplexer performance. Specifically, as s^k increases, the inter-cell correlations present at the source become less dominant and the multiplexer's performance improves accordingly. These observations suggest that, for sufficiently large values of s^k, it may be possible to drop the correlated source model in favor of a simpler, uncorrelated model without significantly impacting on the model's accuracy.

3.1 GPM traffic model for NDP sources

The proposed GPM source can be easily adopted to model such an uncorrelated source with speed-up factor m^k by setting T_s equal to 1 slot; the kth source is then described by the parameters $\{m^k, 1, f_j^k(\cdot)\}$. Note that this is equivalent to setting $T^k = m^k$. Source k is assumed to deliver cells every m^k time slots according to a Bernoulli PMF with rate λ^k. This is captured through the model by setting, for $j = c^k$,

$$f_j^k(i) = \begin{cases} \lambda^k & \text{for } i = 1, \\ 1 - \lambda^k & \text{for } i = 0, \end{cases} \tag{16}$$

and for $j \neq c^k$,

$$f_j^k(i) = \begin{cases} 0 & \text{for } i = 1, \\ 1 & \text{for } i = 0. \end{cases} \tag{17}$$

Note that the analysis presented in Section 2 provides results for a given vector of source start times, while the analyses of [19] and [20] assume random starting times among the sources. For the deterministic (periodic) source models assumed in [19] and [20], random starting times represent the only stochastic element of the resulting queueing system, without which the analysis would be trivial. A more meaningful treatment of the problem, which is adopted here, may be one in which specific – perhaps worst-case – starting times can be assumed. A stochastic cell generation process can then be considered at the deterministic time-scales m^k, which is more representative of realistic traffic generators such as packetized voice and network uplinks with speed mismatches.

3.2. Numerical results for NDP input processes

Although the analysis described in Section 2.4 allows for the superposition of input sources having different periods, the presentation and discussion of the numerical results will be facilitated by considering sources with identical periods denoted by m. Each source is also assumed to deliver cells according to a Bernoulli PMF with rate α, which results in a mean rate per source of α/m; the total offered load is then given by $\lambda = N\alpha/m$.

Fig. 3 illustrates the effect that the source period has on cell-loss probabilities as a function of the buffer capacity B when $N = 40$ sources are multiplexed at an offered load of $\rho = 0.9$ and the sources are assumed to be synchronized ($c^1 = \cdots = c^{40}$). Clearly, the source period m has a significant impact on queue performance when the total offered load is held constant. Note, however, that this impact is not necessarily negative as has been suggested in the past [18]. In fact, for "acceptable" loss probabilities on the order of 10^{-6} and smaller, the impact that increasing values of m have on the queue performance is positive. The curves for m = 10, 20, and 30 are also compared with the corresponding M/D/1/B approximation, and the results are quite enlightening. Fig. 3 reveals that the Poisson assumption can either overestimate or underestimate loss probabilities depending upon buffer capacity and the source period m, which implies that the solution to the M/D/1/B queue should not be used for bounding purposes when considering such periodic input streams. For a loss probability of 10^{-9}, the M/D/1/B approximation is extremely pessimistic, requiring a buffer capacity of $B = 90$, compared to a required buffer capacity of $B = 50$ when $m = 30$. In fact, the negative slope of the M/D/1/B curve is always smaller than the cases for $m > 1$, which implies that, beyond some value of B (which varies with m), the loss probabilities for the limiting Poisson traffic assumption upper bounds the actual loss probabilities for the superposition of non-deterministic periodic sources.

Fig. 4 demonstrates the relationship between cell-loss probabilities and offered load for a case in which the Poisson assumption is optimistic ($B = 30$) as well as a case in which the assumption is overly pessimistic ($B = 50$). The $N = 40$ NDP sources are assumed to have speed-up factors of $m = 30$ associated with them. When $B = 30$, the superposition of NDP input sources results in loss probabilities larger than 10^{-5} for the range of loads between 0.7 and 0.99. The corresponding M/D/1/30 queue, however, is overly optimistic and results in loss probabilities smaller than 10^{-9} for loads as high as 0.72. On the other hand, when $B = 50$, the M/D/1/50 approximation overestimates the loss probabilities for all loads. It is clear from Fig. 4 that the Poisson assumption performs poorly for low to moderate loads, and even for offered loads of 0.99 there is a substantial difference between the M/D/1/B queue performance and the corresponding case of NDP input sources.

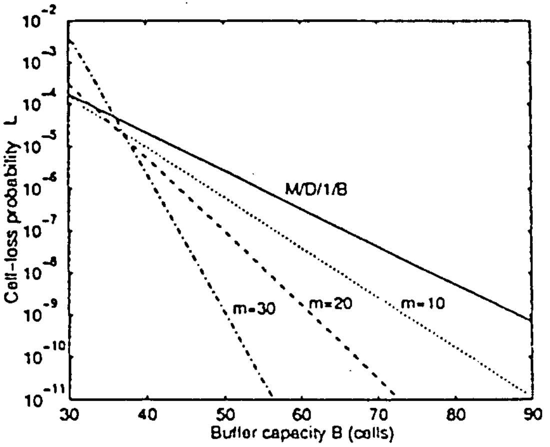


Fig. 3. Loss probability versus buffer capacity B for various m. The number of sources are $N = 40$, and the total load $\lambda = 0.9$.

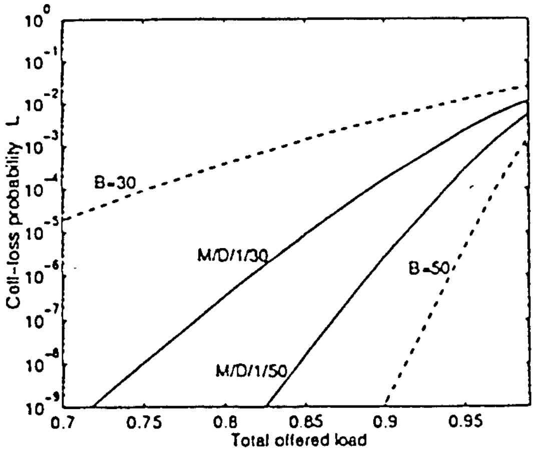


Fig. 4. Loss probability versus offered load for various B. The number of sources are $N = 40$, and speedup factors are set to $m = 30$.

Note that for the results presented in Figs. 3 and 4, it is assumed that the NDP sources are synchronized ($c^1 = \cdots = c^N$), which represents a worst-case scenario in terms of queue performance. Although it is difficult to define a meaningful distribution of initialization times c^k ($1 \leqslant k \leqslant M$) for the NDP sources, it is reasonable to expect that uniformly distributing these times will improve the queueing performance. This is illustrated in Fig. 5, where loss probabilities are plotted versus the number of multiplexed sources with $m = 20$ and $\rho = 0.9$ for synchronized and unsynchronized (initialization times uniformly distributed) sources. The uppermost curves represent results for a buffer capacity of $B = 20$, and the lower curves for $B = 50$. Similarly, the upper star on the right hand vertical axis denotes the M/D/1/20 result while the lower star denotes the M/D/1/50 result. For both cases ($B = 20$ and $B = 50$), the queue performance, in terms of cell-loss probabilities, approaches a limit for large values of N. However, even when the NDP sources are unsynchronized, the limiting results are significantly different than the M/D/1/B results.

The results presented thus far indicate that all of the parameters (B, m, N and λ) affect significantly the queue performance, and consequently, the validity of the M/D/1/B approximation. For each set of numerical results there seem to be two distinct regions of operation; one in which the source periodicities have a negative

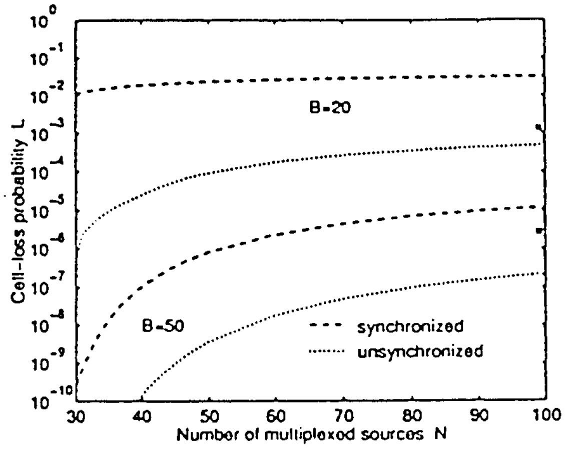


Fig. 5. Loss probability versus N for various B. The total offered load is $\lambda = 0.9$, and speedup factors are set to $m = 20$.

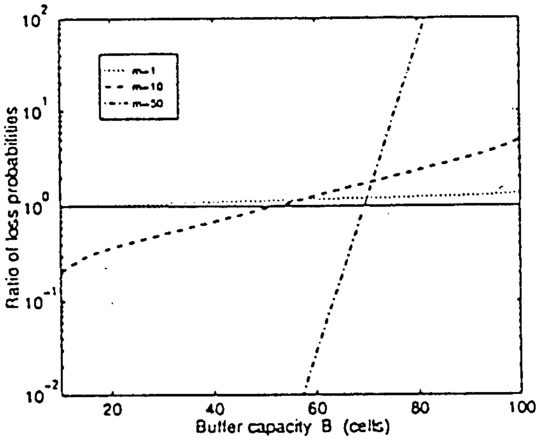


Fig. 6. L_P/L_{NDP} versus B for different values of m. The total offered load is $\lambda = 0.9$, and $N = 75$ sources are multiplexed.

influence on the queueing behavior, relative to a Poisson input source, and one in which the periodicities have a favorable impact. These operating regions are examined more closely in Fig. 6, where the ratio of the loss probability for the Poisson traffic source (L_P) to the loss probability for the NDP sources (L_{NDP}) is plotted versus buffer capacity B. When the ratio L_P/L_{NDP} is less than (greater than) 1, the M/D/1/B approximation underestimates (overestimates) the actual loss probability. In Fig. 6, $N = 75$ non-deterministic periodic sources are multiplexed at a total offered load of 0.9 for source periods of 1, 10 and 50. When $m = 1$, the input to the multiplexer consists of $N = 75$ Bernoulli sources, and the ratio L_P/L_{NDP} is close to 1 for all values of B. This result supports the assertion that the superposition of a large number of Bernoulli traffic streams can be approximated by a Poisson process (asymptotic behavior of the Binomial distribution), regardless of the buffer capacity. However, as the source periods increase, the two regions become more distinct. When $m = 10$ and $B = 10$, L_{NDP} is 5 times greater than L_P for the M/D/1/10 queue, and when $B = 100$, L_{NDP} is 5 times smaller than L_P. For this case, the M/D/1/B results are overly pessimistic for $B \geqslant 50$. When $m = 50$, the two regions are even more well-defined, and the Poisson assumption does not perform well in either region. It is interesting to note that, as was the case in Fig. 3, the region of operating conditions which produces loss probabilities considered acceptable in ATM (less than 10^{-6}) is the one in which the M/D/1/B queue provides overly pessimistic results ($L_P > L_{NDP}$).

4. Application 2: Multiplexing VBR video at slice-level

While the modeling of VBR video sources has recently received significant attention, there presently exists no widely accepted model which lends itself to mathematical analysis. Most existing models, like those in [7] and [8], assume that video is pre-buffered at the frame level. The purpose of pre-buffering is simply to provide

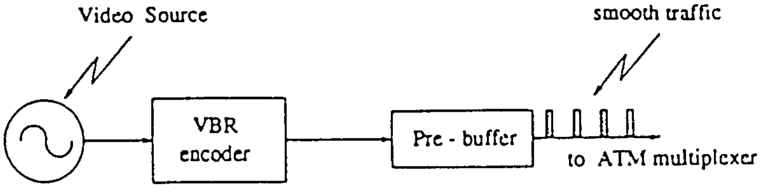


Fig. 7. Pre-buffering a VBR video source.

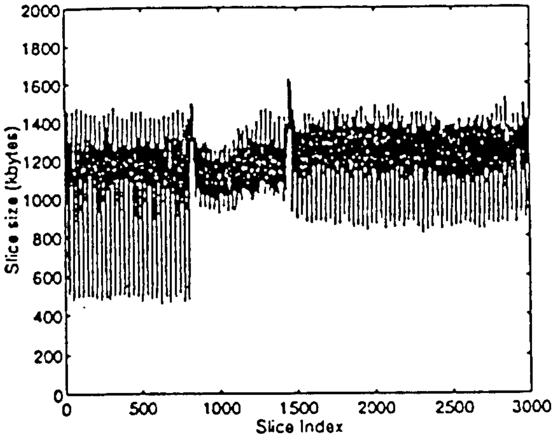


Fig. 8. Slice-level sample of a VBR video sequence.

the network with a cell stream which is as smooth as possible; this can be accomplished by simply storing data temporarily in a pre-buffer as in Fig. 7.

In [11], a flexible simulation model is developed which captures the behavior of VBR video traffic at both the frame and slice level. There are 30 equal-sized slices in every frame, and the inter-frame as well as the inter-slice autocorrelation functions are matched to those from experimental data provided by [12]. The cell generation rate is considered to be constant throughout the slice, modeling the VBR video traffic scheme delivered to the multiplexer under a slice-level pre-buffering scheme. Through a simulation study driven by a number of source models which match the characteristics of a full-motion video movie, [12], the authors show that the impact of inter-frame correlation is negligible while the impact of inter-slice correlation is significant.

These simulation results suggest that if pre-buffering at the frame level is not practical, and transmission takes place at the slice level, then an effective traffic model should capture the time-scales associated with both the frame and the slice. The former will determine the time-scale at which inter-frame correlations exist, while the latter will determine the horizon of the constant-rate cell delivery to the multiplexer (slice). Note that under pre-buffering at the frame level a single rate of the resulting traffic over a frame is generated. Under pre-buffering at the slice level, and for a selected average cell rate over a frame, 30 potentially different and correlated rates of the resulting traffic over a frame are produced; each rate remains fixed over a slice.

Due to the high positive autocorrelations in the number of cells generated in a frame, the cell arrival process (at slice level) over a frame can be considered to remain virtually unchanged over a sequence of consecutive frame horizons. This is in agreement with the experimental data from [12] (Fig. 8) in which the slice-level arrival rate is shown to have a periodic behavior (with period 30 slices), and the frame-level arrival process remains relatively constant over many frames (or periods). Assuming that the time horizon associated with changes in the frame-level arrival rate is sufficiently long to induce the steady-state behavior of the multiplexer, a quasi-static approximation similar to that in [9] can be used to study the multiplexing of VBR traffic which is pre-buffered at the slice level as described above. However, a constant-rate arrival process over a frame, such as the Poisson process considered in [9], will clearly not be applicable here to approximate the arrival process over a number of video frames. Instead, assuming a frame-level arrival rate which remains constant over many frames, a periodic cell arrival process (with a period equal to the frame length) within each frame could be considered.

4.1. GPM traffic model for the periodic slice-level arrival process

The periodic cell arrival process described above can be easily captured through the proposed GPM traffic model by setting the period of each GPM source (m^k) to be 30 and the sub-period (T_s) equal to the number of

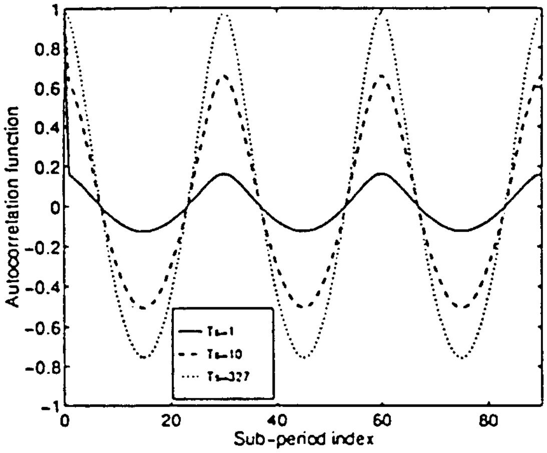


Fig. 9. Sub-period-level autocorrelation function for $\lambda^k = 0.2$, $b = 0.2$ and various values of T_s.

slots in a slice; T_s varies with the link speed, the frame rate (in frames/sec) and the slot capacity. Assuming a frame rate of 24 per second, as in [11] and [12], and a slot capacity of 53 bytes (48 bytes of payload and 5 bytes of header) for ATM,

$$T_s = \frac{C \text{ bits}}{\text{sec}} \cdot \frac{1 \text{ byte}}{8 \text{ bits}} \cdot \frac{1 \text{ slot}}{53 \text{ bytes}} \cdot \frac{1 \text{ frame}}{30 \text{ slices}} \cdot \frac{1 \text{ sec}}{24 \text{ frames}}$$
$$= \frac{C}{305{,}280} \text{ slots/slice,} \tag{18}$$

where C is the link capacity in units of bits/sec. To capture the uniform distribution of cells within a slice, the PMF $f_j^k(\cdot)$ for the number of cells generated in each slot when source k is in state j ($1 \leqslant j \leqslant 30$) will be Bernoulli with rate λ_j^k. Note that this can be seen as statistically smoothing the source at the slice level. It is believed that an assumption of statistical smoothing should provide for a conservative (or pessimistic) estimation of delivered QoS if the VBR source were deterministically smoothed (cells are delivered to the network equi-spaced within each slice). This has been shown to be the case in [23] when the VBR source is pre-buffered and smoothed on a frame by frame basis.

As was the case with the slice-level arrival rate process, the slice-level autocorrelation function for a VBR video source has been shown [10,12,13] to possess a periodic structure with a period of 30 slices. The GPM traffic model also has a periodic sub-period-level autocorrelation function $R^k(n)$, as given in Eq. (4). In fact, for a given frame-level cell arrival rate for source k, denoted by λ^k in Eq. (1), the slice-level autocorrelation function for the kth video source can be approximated by $R^k(n)$ through an appropriate selection of the arrival rates λ_j^k, $1 \leqslant j \leqslant 30$.

In order to produce a first approximation for the autocorrelation function for cells/slice which is similar to the functions found in [24], for example, the following distribution of the rates λ_j^k will be considered [3]:

$$\lambda_i^k = \lambda_{(31-i)}^k = a e^{-(i-1)b}, \quad i = 1, 2, \dots, 15. \tag{19}$$

[3] Note that there exist many possible choices for the rate distribution function λ_i^k. The distribution given in Eq. (19) is intended only to serve as an example, and is not intended to closely match any experimental data set.

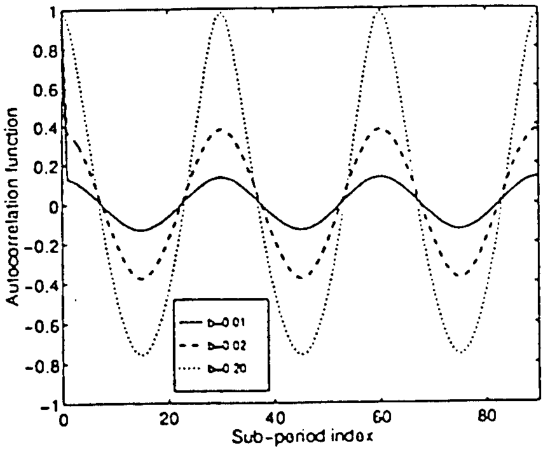


Fig. 10. Sub-period-level autocorrelation function for $\lambda^k = 0.2$, $T_s = 327$ and various values of b.

Upon choosing a value of b to produce the desired profile for the autocorrelation function, the constant a is calculated using the expression

$$\lambda^k = \frac{a}{15}\sum_{i=1}^{15} e^{-(i-1)b}, \tag{20}$$

which is derived from (1) by requiring that the mean rate over states 1 through 15 (which equals the mean rate over states 16 through 30) be equal to the frame cell rate λ^k for the source. Note that according to (4), the autocorrelation function for cells/sub-period is dependent upon both T_s and the sub-period rates λ_j^k. Fig. 9 shows the resulting sub-period-level autocorrelation functions for a source with frame cell rate $\lambda^k = 0.2$, $b = 0.2$ and different values of T_s ranging from 1 to 327. The latter corresponds to a video source being multiplexed onto a 100 Mbps ATM link (see Eq. (18)).

Similarly, Fig. 10 presents sub-period-level autocorrelation functions for the source with $T_s = 327$ and different sub-period rate distributions resulting by varying b from 0.01 to 0.20. Note that as b approaches 0, the GPM source looks more and more like an i.i.d. cell arrival process. In fact, when $b = 0$, the GPM source is simply a binomially distributed batch arrival process. It is clear from Figs. 9 and 10 that similar sub-period-level autocorrelation functions can be produced by either fixing b and varying T_s, or fixing T_s and varying b.

4.2. Numerical results and discussion for the VBR application

In this section some numerical results are presented for the analysis of the finite capacity multiplexer with an input process consisting of the superposition of N GPM traffic sources, which are adopted as explained in the previous section to model VBR video sources pre-buffered at the slice level. The stationary queue distributions computed using the techniques presented in Section 2.4 yield queueing results for video sources generating traffic according to a single average frame rate. In order to compute the queue length distributions over all possible frame rates, a frame rate histogram can be derived as described in [9], and the quasi-static approximation presented there can be applied. Application of the quasi-static approximation is beyond the scope of this paper. The primary objective here is to determine the impact of periodic slice-level autocorrelations on queue response.

To address the effectiveness of the slice-level autocorrelation function in describing a VBR input source, consider the functions pictured in Fig. 11. Although the two autocorrelation functions are almost identical, one is produced by an underlying periodic Markov chain which makes transitions every time slot ($T_s = 1$), while the

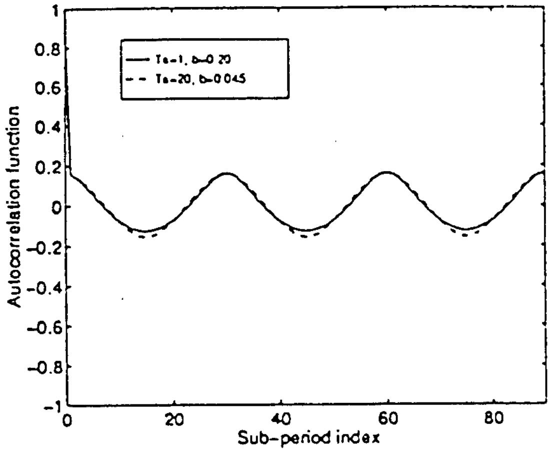


Fig. 11. Sub-period-level autocorrelation functions for sources with $\lambda^{t} = 0.2$, and different values of T_s.

other is produced by a source making transitions every 20 slots ($T_s = 20$). The autocorrelation functions are matched by simply varying the distribution of sub-period rates through the parameter b; the total rate for each GPM source is $\lambda^{t} = 0.2$. The queue response to these sources is examined by considering a multiplexer of capacity $B = 30$ cells with an input of 4 of the GPM sources. The input sources with $T_s = 20$ slots have a much more negative impact on the multiplexer than those with $T_s = 1$, which is illustrated in Fig. 12 through tail distributions for buffer occupancy as well as loss probabilities L. It is clear that simply matching the autocorrelation function for cells/slice is not sufficient to estimate multiplexer performance. The actual time-scale associated with the slice needs to be considered carefully.

In fact, by adopting different source time-scales T_s, performance studies can lead to contrasting conclusions about the feasibility of multiplexing VBR video which is not pre-buffered at the frame level. For instance, based on numerical results for input sources with periodic autocorrelations which make transitions every time slot, the author in [25] concludes that loss probabilities are higher when VBR video sources are pre-buffered and smoothed on a frame by frame basis. A similar conclusion would be drawn using GPM sources with $T_s = 1$. For

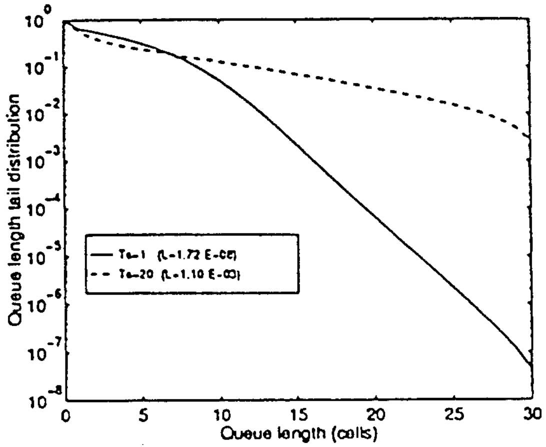


Fig. 12. Tail distributions and loss probabilities for matching autocorrelation functions with $T_s = 1$ and 20; total load is $\lambda = 0.8$ in both cases.

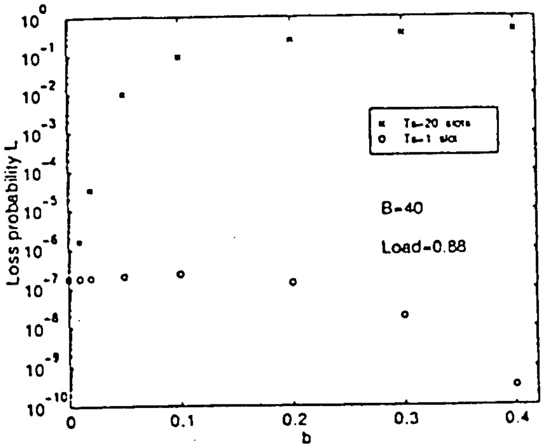


Fig. 13. Cell loss probabilities for $N = 4$ GPM sources, $B = 40$ and T_s of 1 and 20; total load is $\lambda = 0.88$.

GPM input sources, a frame-level pre-buffering scheme is modeled by setting constant arrival rates over all 30 sub-periods (or slices), which is the case when $b = 0$ under the rate distribution strategy of Eq. (19). Note that this eliminates the periodic nature of the slice-level autocorrelation function.

In Fig. 13 cell loss probabilities are plotted versus b for $N = 4$ input sources, each with rate 0.22, at a multiplexer of capacity $B = 40$ for $T_s = 1$ and $T_s = 20$. As expected, when $b = 0$ (which models frame-level pre-buffering) the rates do not change from slice to slice and the loss probability does not depend upon T_s. When $T_s = 1$, notice that loss probabilities decrease as the periodic fluctuations in arrival rate increase with b. Therefore, use of this model would lead to the false conclusion, as in [25], that the periodic autocorrelations provide for improved statistical multiplexing gain over frame-level pre-buffering. As T_s is increased to as little as 20 slots, which corresponds to an output link of capacity $C = 6.1$ Mbps (Eq. (18)) for the adopted GPM model, numerical results for L lead to a completely different conclusion. In this case it would seem that smoothing the periodic autocorrelations through frame-level pre-buffering provides greatly improved QoS. These results illustrate the importance of incorporating time-scales into traffic models in order to accurately assess the impact of the modeled traffic sources on network performance.

In order to investigate a more realistic multiplexing scenario, Table 1 depicts cell loss probabilities for a buffer capacity of $B = 50$, an output link capacity of 100 Mbps, and an input consisting of 10 identical VBR sources each transmitting at a rate of 8 Mbps. This scenario can be modeled with GPM sources through the following choice of parameters: $T_s = 327$, $\lambda^1 = \cdots = \lambda^{10} = 0.08$ and of course $m^1 = \cdots = m^{10} = 30$. As explained in Section 4.1, each source generates cells according to the appropriate Bernoulli PMF to approximate the smoothing of cells within a slice.

In Table 1, three different autocorrelation functions for cells/slice are considered. One set of results represents the frame-level pre-buffering case ($b = 0$) while the others correspond to two of the autocorrelation

Table 1
Loss probabilities for different values of b under both *synchronized* and *uniformly distributed* frame boundaries

b	L for *synchronized* frame boundaries	L for *uniformly distributed* frame boundaries
0.00	6.57 e^{-12}	6.57 e^{-12}
0.10	7.43 e^{-10}	6.54 e^{-12}
0.20	4.67 e^{-7}	6.45 e^{-12}

curves pictured in Fig. 10 ($b = 0.01$ and $b = 0.02$). Clearly, the periodic autocorrelation function has a negative impact on the queue, and increasing the positive and negative peaks in the autocorrelation function by increasing b results in even worse performance. Note that the worst-case scenario in terms of possible starting times among the input sources is achieved when the frame boundaries of all 10 sources are synchronized ($c^1 = \cdots = c^{10}$).

The impact of frame-boundary synchronization can be minimized by uniformly distributing the frame starts (boundaries) at intervals of $\lfloor M/N \rfloor$, as is the case in [11], which represents a best-case scenario in terms of source starting times. While ensuring such a uniform distribution of frame starts in practice is highly unlikely, the positive impact on the multiplexer is indisputable, as is seen in Table 1, where loss probabilities virtually coincide for all values of b.

Recall that the results presented here are for one possible combination of frame cell rates, and that under a quasi-static approximation, all combinations of the individual frame cell rates would be considered. However, it should be noted that the averaging effect of the quasi-static approximation may mask the extreme queueing conditions that occur over time horizons for which the offered load is high. Hence, it may be more reasonable, when applying the quasi-static approximation, to consider only the "high-rate" bins, which would yield a more detailed account of the queueing behavior during the times in which the QoS requirements for the VBR sources are most likely to be violated. It is possible that these QoS violations, which may be sufficiently rare when considered over all possible combinations of frame cell rates, are severe enough during the high-load periods to disrupt, or at least impair, service.

5. Conclusions

In this paper, the concept of source time-scales was applied to the discussion of traffic sources in broadband packet networks such as ATM. A Generalized Periodic Markovian (GPM) traffic model was introduced to capture the cell arrival dynamics associated with these time-scales, and a finite-capacity multiplexer with an input process consisting of the superposition of N GPM sources was analyzed in discrete time. An efficient numerical technique was presented for the solution of a potentially large irreducible Markov chain by solving smaller chains with state space size independent of the number of states in the input process.

As expected, numerical results suggest that the time-scales at which variations in the cell arrival process occur, impact significantly on the queue response. This fact seems to indicate that these relevant time-scales, which change with the link speed, should be considered as important traffic descriptors, and incorporated into source models in high-speed environments.

The proposed GPM traffic source model and developed analytical technique were applied to the study of Non-Deterministic Periodic (NDP) traffic sources as well as to VBR video sources which deliver cells at the slice level, as opposed to smoothing and delivering them on a frame by frame basis. Results from the NDP source study indicate that, depending upon the operating conditions at the finite-capacity multiplexer, the period of the NDP input streams may have either a positive or negative impact on queueing performance, relative to that of a M/D/1/B queue. This implies that M/D/1/B results should not be used for bounding loss probabilities when studying the queue response to NDP input processes. Furthermore, the asymptotic (large N) behavior of NDP input sources was shown to be poorly approximated by the Poisson process when the potential cell generation times occur periodically with period greater than 1 (i.e. $m > 1$).

Results from the VBR video application illustrate that the inter-slice autocorrelation function, which is periodic in nature, has a major impact on the finite-capacity queue. The video application was based on the assumption, first made in [9], that the system can be approximately analyzed by independently studying the queue response to different frame cell rates and then conditioning over all rates. Numerical results, which were derived for a single frame cell rate, indicate that frame-level pre-buffering provides improved QoS when slice-level autocorrelations are periodic.

Appendix A. Derivation of the sub-period-level autocorrelation function $R(n)$

This appendix is concerned with the derivation of $R(n)$, defined as

$$R(n) \triangleq \frac{\mathrm{E}\{A_{n'}A_{n'+n}\} - \mathrm{E}^2\{A_{n'}\}}{\mathrm{E}\{A_{n'}^2\} - \mathrm{E}^2\{A_{n'}\}}. \tag{A.1}$$

The quantity $\mathrm{E}\{A_{n'}A_{n'+n}\}$ will be derived by considering the cases $n = 0$ and $n > 0$ separately. For $n = 0$,

$$\begin{aligned} \mathrm{E}\{A_{n'}^2\} &= \sum_k k^2 \Pr\{A_{n'} = k\} \\ &= \sum_i \sum_k k^2 \Pr\{A_{n'} = k \mid \gamma_{n'} = \lambda_i\} \Pr\{\gamma_{n'} = \lambda_i\}. \end{aligned} \tag{A.2}$$

The second moment of $A_{n'}$, given $\gamma_{n'}$, will be denoted by ξ_2, where

$$\xi_2 = \sum_k k^2 \Pr\{A_{n'} = k \mid \gamma_{n'} = \lambda_i\}. \tag{A.3}$$

This quantity can be expressed in terms of the GPM parameters defined earlier once the cell arrival PMF $f_i(\cdot)$, $1 \leqslant i \leqslant m$, is known. For instance, if $f_i(\cdot)$ is a Bernoulli PMF with rate λ_i, then $\Pr\{A_{n'} = k \mid \gamma_{n'} = \lambda_i\}$ is simply a binomial distribution given by

$$\Pr\{A_{n'} = k \mid \gamma_{n'} = \lambda_i\} = \binom{T_s}{k} (\lambda_i)^k (1 - \lambda_i)^{T_s - k}. \tag{A.4}$$

In this case, it is easy to show that ξ_2 is simply

$$\xi_2 = T_s^2 \lambda_i^2 + T_s \lambda_i (1 - \lambda_i), \tag{A.5}$$

and (22) can be expressed as

$$\begin{aligned} \mathrm{E}\{A_{n'}^2\} &= \sum_i \left[T_s^2 \lambda_i^2 + T_s \lambda_i (1 - \lambda_i)\right] \pi_i \\ &= T_s^2 \sum_i \lambda_i^2 \pi_i + T_s \lambda - T_s \sum_i \lambda_i^2 \pi_i. \end{aligned} \tag{A.6}$$

For $n > 0$,

$$\begin{aligned} \mathrm{E}\{A_{n'}A_{n'+n}\} &= \sum_{k_1} \sum_{k_2} k_1 k_2 \Pr\{A_{n'} = k_1,\ A_{n'+n} = k_2\} \\ &= \sum_i \sum_j \sum_{k_1} \sum_{k_2} k_1 k_2 \Pr\{A_{n'} = k_1 \mid \gamma_{n'} = \lambda_i\} \Pr\{A_{n'+n} = k_2 \mid \gamma_{n'+n} = \lambda_j\} \\ &\qquad \cdot \Pr\{\gamma_{n'+n} = \lambda_j \mid \gamma_{n'} \lambda_i\} \Pr\{\gamma_{n'} = \lambda_i\} \\ &= T_s^2 \sum_i \sum_j \lambda_i \lambda_j p_{i,j}^{(n)} \pi_i. \end{aligned} \tag{A.7}$$

From (A.6) and (A.7), the quantity $\mathrm{E}\{A_{n'}A_{n'+n}\}$, for $n \geqslant 0$, can be expressed as

$$\mathrm{E}\{A_{n'}A_{n'+n}\} = T_s^2 \sum_i \sum_j \lambda_i \lambda_j p_{i,j}^{(n)} \pi_i + T_s \left(\lambda - \sum_i \lambda_i^2 \pi_i\right) \delta_n, \tag{A.8}$$

where δ_n is the Kronecker delta.

This expression can be further simplified by noting that, due to the periodic structure of the Markov chain considered here, the stationary probability π_i is simply $1/m$, where m represents the number of states in the periodic Markov chain. Also note that the one-step transition probabilities for this source are defined by

$$p_{i,j}=\begin{cases}1 & \text{if } 1\leqslant i\leqslant m \text{ and } j=(i) \bmod (m)+1,\\ 0 & \text{otherwise.}\end{cases} \tag{A.9}$$

From these quantities it is an easy computation to show that the n-step transition probabilities are given by

$$p_{i,j}^{(n)}=\begin{cases}1 & \text{if } 1\leqslant i\leqslant m \text{ and } j=(i+n-1) \bmod (m+1),\\ 0 & \text{otherwise.}\end{cases} \tag{A.10}$$

Eq. (A.8) can now be rewritten as

$$\mathrm{E}\{A_{n'}A_{n'+n}\}=\frac{T_s^2}{m}\sum_{i=1}^{m}\lambda_i\lambda_j+T_s\left(\lambda-\frac{1}{m}\sum_{i=1}^{m}\lambda_i^2\right)\delta_n, \tag{A.11}$$

where $j=(i+n-1)\bmod(m)+1$.

Finally, $\mathrm{E}\{A_{n'}\}$ can be derived as follows:

$$\begin{aligned}\mathrm{E}\{A_{n'}\}&=\sum_k k\Pr\{A_{n'}=k\}\\ &=\sum_i\sum_k k\Pr\{A_{n'}=k\mid\gamma_{n'}=\lambda_i\}\Pr\{\gamma_{n'}=\lambda_i\}\\ &=\sum_i T_s\lambda_i\pi_i\\ &=T_s\lambda.\end{aligned} \tag{A.12}$$

By substituting (A.11) and (A.12) into (A.1), the final expression for the autocorrelation function $R(n)$ is obtained:

$$R(n)=\frac{\dfrac{1}{m}\displaystyle\sum_{i=1}^{m}\lambda_i\lambda_j+\frac{1}{T_s}\left(\lambda-\frac{1}{m}\sum_{i=1}^{m}\lambda_i^2\right)\delta_n-\lambda^2}{\dfrac{1}{m}\displaystyle\sum_{i=1}^{m}\lambda_i^2+\frac{1}{T_s}\left(\lambda-\frac{1}{m}\sum_{i=1}^{m}\lambda_i^2\right)-\lambda^2}, \tag{A.13}$$

where again $j=(i+n-1)\bmod(m)+1$.

Appendix B. Derivation of the quantities $\hat{\pi}_{i,j}$ and L

Let X_n denote the process $\{q_n, x_n\}_{n\geq 0}$, where n is indexed at *slot* boundaries. Recall that the process $\{q_k, x_k\}_{k\geq 1}$, which has been considered for the analysis of the GPM sources at a finite-capacity multiplexer, is indexed at sub-period boundaries; the length of a sub-period is equal to T_s slots. The Markov chain $\{q_k, x_k\}_{k\geq 1}$ has state-space $S=\{(0, 1),\dots,(B, M)\}$. Note that $\{q_k, x_k\}_{k\geq 1}$ is embedded in the process $\{q_n, x_n\}_{n\geq 0}$, and $\{X_{(k-1)T_s}\}_{k\geq 1}=\{q_k, x_k\}_{k\geq 1}$.

This appendix is concerned with the computation of the stationary probabilities for the process X_n, denoted by $\hat{\pi}_{i,j}$ where

$$\hat{\pi}_{i,j}=\lim_{n\to\infty}\Pr\{q_n=i,\ x_n=j\},$$

as well as the cell loss probability L.

Let ϕ denote some event of interest which can be observed at slot boundaries. $\Pr\{\phi\}$ will denote the probability of this event. Let $N_{i,j}(\phi)$ denote the number of occurrences of ϕ when the Markov chain $\{q_k, x_k\}_{k \geq 1}$ is in state $(i, j) \in S$. $\bar{N}_{i,j}(\phi)$ will denote the expected value of this quantity. If $\pi_{i,j}$ denotes the stationary probability that the Markov chain $\{q_k, x_k\}_{k \geq 1}$ is in state $(i, j) \in S$, then it can be shown [26] that

$$\Pr\{\phi\} = \frac{1}{T_s} \sum_i \sum_j \bar{N}_{i,j}(\phi) \pi_{i,j}. \tag{B.1}$$

Computation of $\hat{\pi}_{i,j}$. Let ϕ represent the event that the system is found in state (i, j) at an arbitrary time instant, and let $N_{i',j'}(i, j)$ denote the number of times the system is in state (i, j) over the current sub-period, given that the sub-period began in state (i', j'). Also denote by $I^n_{i',j'}(i, j)$ an indicator function which takes the value 1 if the system is in state (i, j) after n time slots in the sub-period which began in state (i', j'); the quantity equals 0 otherwise. Clearly, $N_{i',j'}(i, j)$ can be expressed as

$$N_{i',j'}(i, j) = \sum_{n=1}^{T_s} I^n_{i',j'}(i, j).$$

The expected value of $N_{i',j'}(i, j)$ is given by

$$\begin{aligned}\bar{N}_{i',j'}(i, j) &= \sum_{n=1}^{T_s} \mathrm{E}\{I^n_{i',j'}(i, j)\}\\ &= \sum_{n=1}^{T_s} \Pr\{q_n = i, x_n = j \mid q_0 = i', x_0 = j'\}\\ &= \sum_{n=1}^{T_s} p_j^{(n)}(i', i) 1_{\{j=j'\}}.\end{aligned}$$

Finally, a direct application of Eq. (B.1) yields the result

$$\hat{\pi}_{i,j} = \frac{1}{T_s} \sum_{n=1}^{T_s} \sum_{i'=0}^{B} p_j^{(n)}(i', i) \pi_{i',j} \quad \text{for}(i, j) \in S. \tag{B.2}$$

Computation of L. In this case, ϕ is the event of a cell being lost in an arbitrary time slot. By definition, the loss rate, or cell loss probability L, is given by

$$L = \frac{\Pr\{\phi\}}{\lambda},$$

where λ is the total offered load, or the probability that a cell arrives in an arbitrary time slot. $N_{i'j'}$ will denote the total number of cells lost over the sub-period which began in state (i', j'); $\bar{N}_{i'j'}$ will denote its expected value. Let $l^n_{i'j'}$ denote the number of cells lost over the nth slot of this sub-period and let a_n denote the number of cell arrivals over the nth slot. It is easy to see that

$$l^n_{i'j'} = \begin{cases} k & \text{if } a_n = B + 1 - q_{n-1} + k, \\ 0 & \text{otherwise.} \end{cases}$$

Clearly,

$$N_{i'j'} = \sum_{n=1}^{T_s} l^n_{i'j'}.$$

and

$$\begin{aligned}\bar{N}_{i'j'} &= \sum_{n=1}^{T_s} \mathrm{E}\{I_{i'j'}^n\}\\ &= \sum_{n=1}^{T_s} \sum_k k \Pr\{a_n = B+1-q_{n-1}+k \mid q_0 = i',\ x_0 = j'\}\\ &= \sum_{n=1}^{T_s} \sum_{i=0}^{B} \sum_k k \Pr\{a_n = B+1-i+k \mid q_0 = i',\ x_0 = j'\} \Pr\{q_{n-1} = i \mid q_0 = i',\ x_0 = j'\}\\ &= \sum_{n=1}^{T_s} \sum_{i=0}^{B} \sum_{k=K_1}^{K_2} k p_{j'}^{(n-1)}(i',\ i) \bar{f}_{j'}(B+1-i+k),\end{aligned}$$

where K_1 is the minimum number of losses which can occur over the given time slot, and K_2 is the maximum number of such losses. It is easy to see that $K_1 = \max(0,\ i-B-1)$ and $K_2 = N-B-1+i$.

Finally, applying Eq. (B.1) yields

$$L = \frac{1}{\lambda T_s} \sum_{i'=0}^{B} \sum_{j'=1}^{M} \sum_{n=1}^{T_s} \sum_{i=I_1}^{B} \sum_{k=K_1}^{K_2} k p_{j'}^{(n-1)}(i',\ i) \bar{f}_{j'}(B+1-i+k) \pi_{i',j'}, \qquad \text{(B.3)}$$

where $I_1 = \max(0,\ B+2-N)$.

References

[1] H. Heffes and D. Lucantoni, A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer analysis, *IEEE J. Select. Areas Commun.* (September 1986).

[2] K. Sriram and W. Whitt, Characterizing superposition arrival processes in packet multiplexers for voice and data, *IEEE J. Select Areas Commun.* (September 1986).

[3] J. Daigle, Y. Lee and N. Magalhaes, Discrete time queues with phase dependent arrivals, in: *Proc. IEEE INFOCOM'90*, San Francisco, CA, 1990.

[4] I. Stavrakakis, Analysis of a statistical multiplexer under a general input traffic model in: *Proc. IEEE INFOCOM'90*, San Francisco, CA, 1990.

[5] S.Q. Li, A general solution technique for discrete queueing analysis of multimedia traffic on ATM, *IEEE Trans. Commun.* 39 (7) (1991).

[6] K. Sohraby, On the asymptotic behavior of heterogeneous statistical multiplexer with applications, in: *Proc. IEEE INFOCOM'92*, Florence, Italy, 1992.

[7] B. Maglaris, D. Anastassiou, S. Prodip, G. Karlsson and J. Robbins, Performance models of statistical multiplexing in packet video communications, *IEEE Trans. Commun.* 36 (July 1988).

[8] C. Blondia and O. Casals, Performance analysis of statistical multiplexing of VBR sources, in: *Proc. IEEE INFOCOM'92*, Florence, Italy, 1992.

[9] P. Skelly, M. Schwartz and S. Dixit, A histogram-based model for video traffic behavior in an atm multiplexer, *IEEE/ACM Trans. Networking* 1 (4) (1993).

[10] P. Pancha and M. El Zarki, MPEG coding for variable bit rate video transmission, *IEEE Commun. Mag.* 32 (5) (1994).

[11] A. Lazar, G. Pacifici and D. Pendarakis, Modeling video sources for real-time scheduling, in: *Proc. IEEE GLOBECOM'93*, Houston, TX, 1993.

[12] M.W. Garrett and M. Vetterli, Congestion control strategies for packet video, in: *Proc. 4th Internat. Workshop on Packet Video*, Kyoto, Japan, 1991.

[13] B. Melamed, D. Raychaudhuri, B. Sengupta and J. Zdepski, Tes-based traffic modeling for performance evaluation of integrated networks, in: *Proc. IEEE INFOCOM'92*, Florence, Italy, 1992.

[14] S.Q. Li and C.L. Hwang, Queue response to input correlation functions: Discrete spectral analysis, in: *Proc. IEEE INFOCOM'92*, Florence, Italy, 1992.

[15] J. Hunter, *Mathematical Techniques of Applied Probability, Discrete Time Models: Basic Theory*, Volume 1 (Academic Press, New York, 1983).

[16] J. Hunter, *Mathematical Techniques of Applied Probability, Discrete Time Models: Techniques and Applications*, Volume 2 (Academic Press, New York, 1983).

[17] V. Ramaswami and G. Latouche, Modeling packet arrivals from asynchronous input lines, in: *Proc. ITC 12*, 1988.

[18] T. Eliazov, V. Ramaswami, W. Willinger and G. Latouche, Performance of an ATM switch: Simulation study, in: *Proc. IEEE INFOCOM'90*, San Francisco, CA, 1990.

[19] A. Eckberg, The single server queue with periodic arrival process and deterministic service time, *IEEE Trans. Commun.* 27 (March 1979).

[20] J. Roberts and J. Virtamo, The superposition of periodic cell arrival streams in an ATM multiplexer, *IEEE Trans. Commun.* 39 (2) (1991).

[21] H. Kroner, Statistical multiplexing of sporadic sources-exact and approximate performance analysis, in: *Teletraffic and Datatraffic in a Period of Change: ITC 13* (Elsevier, Amsterdam, 1991).

[22] I. Stavrakakis, Statistical multiplexing of correlated slow traffic sources, in: *Proc. GLOBECOM'92*, Orlando, FL, 1992.

[23] N. Shroff and M. Schwartz, Video modeling within networks using deterministic smoothing at the source, in: *Proc. IEEE INFOCOM'94*, Toronto, Canada, 1994.

[24] B. DeCleene, P. Panch, M. El Zarki and H. Sorensen, Comparison of priority partition methods for VBR MPEG, in: *Proc. IEEE INFOCOM'94*, Toronto, Canada, 1994.

[25] C. Herrmann, VBR video in ATM without frame-buffering: Influence of a periodic correlation function on QoS parameters, in: *Proc. 2nd Workshop on Performance Modelling and Evaluation of ATM Networks*, Bradford, UK, 1994.

[26] H.C. Tijms, *Stochastic Modeling and Analysis: A Computational Approach* (Wiley, New York, 1986).

Randall Landry received his B.S. degree in Electrical Engineering from the University of Southern Maine in 1990 and his M.S. and Ph.D. degrees in Elecrical Engineering from the University of Vermont in 1992 and 1994.

He is currently employed in the Core Network Technology Branch of the DSP R&D Center at Texas Instruments, Inc., Dallas, Texas. His research interests include queueing theory, performance analysis and design of communication systems, and modeling of broadband integrated services traffic.

Dr. Landry is currently engaged in research efforts aimed at developing and optimizing technologies for the internetworking of data communication protocols such as Ethernet and ATM. His work in this area encompasses protocol, algorithm and hardware development. He is also involved in efforts to develop very high-speed switch architectures and traffic management protocols.

Ioannis Stavrakakis received the Diploma in Electrical Engineering from the Aristotelian University of Thessaloniki, Thessaloniki, Greece, 1983, and the Ph.D. degree in Electrical Engineering from the University of Virginia, 1988.

In 1988, he joined the faculty of Computer Science and Electrical Engineering at the University of Vermont as an assistant and then associate professor. Since 1994, he has been an associate professor of Electrical and Computer Engineering at Northeastern University, Boston. His research interests are in stochastic system modeling, teletraffic analysis and discrete-time queueing theory, with primary focus on the design and performance evaluation of Broadband Integrated Services Digital Networks (B-ISDN).

Dr. Stavrakakis is a senior member of IEEE and a member of the IEEE communications Society, Technical Committee on Computer Communications. He has organized and chaired sessions, and has been a technical committee member, for conference such as GLOBECOM, ICC and INFOCOM.

ELSEVIER

Computer Communications 19 (1996) 1112-1123

computer communications

Characterization and multiplexing of correlated traffic from sources with different time constants

Ioannis Stavrakakis*

Electrical and Computer Engineering, 409 Dana Building, Northeastern University, Boston, MA 02115, USA

Abstract

Single server, discrete-time queueing systems under independent or Markov arrival processes and deterministic service requirements have been studied widely in the past. The fixed service time is usually set equal to one (the network-time-constant), defining the time unit of the discrete-time system axis. Such queueing systems have been adopted for the modeling of statistical multiplexers in slotted communication networks supporting sources of (fixed size) packetized information. In this paper, a statistical multiplexer fed by Markov packet sources is considered. Unlike previous considerations, the time-constant of the sources is assumed to be greater than one. As a result, packets cannot be generated over consecutive discrete-time instants whose distance is less than the source time-constant, although the source may be active; both the cases of Markov sources with single and double time-constants are considered. The resulting packet traffic is characterized and the formulated queueing model is studied. In addition to presenting an analysis of a statistical multiplexer fed by quite general Markov traffic streams, this study concludes that the negative impact of source traffic correlations on statistical multiplexing can be reduced substantially when sources with time-constants greater than one are considered. Even in the absence of correlations, the guaranteed idleness of the source within a source-time-constant's interval is shown to have a positive impact on the multiplexing process. Finally, some numerical results for the case of multiplexed packetized voice sources are also presented.

Keywords: Statistical multiplexing; Time constraints; Markov traffic streams

1. Introduction

The probabilistic nature of the traffic generated by users of a networking facility renders inefficient the exclusive allocation of the resources to individual users over a certain time horizon. Some type of statistical multiplexing of the information generated by different sources is usually employed at the access points of a network. In this paper N sources of (fixed size) packetized information are multiplexed before accessing a fixed speed, slotted transmission line (Fig. 1). The constant packet transmission time (slot) is assumed to be equal to one and is called the network time-constant or the network (output) time slot. The slot boundaries of the output transmission line define the discrete-time axis of the system. Packets are temporarily stored in a buffer until the output transmission line is available. Packet arrivals and departures are assumed to occur at the discrete-time instants.

The above statistical multiplexer has been studied in the past under correlated packet arrival processes. The most relevant work can be found elsewhere [1-10] (and the references cited there). In Heffes and Lucantoni [1], the aggregate traffic generated by multiple independent input lines is modeled as a 2-state Markov Modulated Poisson process using moment matching techniques. Then, the resulting statistical multiplexer is studied. In Towsley [2], the author considers a single input line and arrivals which depend on an underlying two state Markov chain. In Bruneel [3], Viterbi [4], and Reiser [5], N input lines are assumed present. In Bruneel [3], it is assumed that the packet arrival process in each of the identical input lines depends on an underlying two state Markov chain (active/inactive). In Viterbi [4] it is assumed that the per line packet arrival process is a first order Markov chain, and at most one packet arrival is possible. A closed form solution for the mean packet delay has been derived for the latter case. In Reiser [5], a closed form expression for the mean packet delay in the case of Bernoulli per line arrivals can be found. The

[*] Email: ioannis@cdsp.neu.edu

PII: S0140-3664(96)01137-1

discrete time system considered in Ali Khan [6, 7] corresponds to a statistical multiplexer where packets arrive through a single input line and they form a Markov chain. Finally, statistical multiplexers under discrete-time phase-type arrival processes have been considered [8–10]. More references on queueing models for statistical multiplexers may be found in Daigle [11].

Extensive previous work on similar queueing systems in a continuous time set up can be also found in the literature. In Anick et al. [12], N identical on-off sources with exponentially distributed on and off periods are considered. The case of N identical input lines with exponentially distributed idle periods and arbitrarily distributed active periods is studied in Refs. [13, 14]. A similar model when the input lines are not identical has been considered [15]. Other continuous time models in a multi-entry buffer system have been considered [16, 17]. Finally, when the Markov processes describing the per input line packet arrivals is a reversible process, a decomposition method that circumvents the state explosion problem has been presented [18].

The correlated (Markov dependent) discrete-time packet arrival processes considered in the past is typically assumed to be generated by correlated sources whose time-constant is equal to the network time-constant; the source time constant is defined as the minimum time interval between consecutive packet generation instants. In this paper, the packet sources are assumed to be correlated, as determined by a first-order Markov model. Unlike most previous considerations, the transition time of the Markov model (source time-constant) is greater than the packet transmission time (network time-constant). The resulting traffic is more complex and defines a queueing system which is different from those considered in the past under correlated traffic. A relevant queueing system has been studied [19, 20] under non-Markov correlated arrivals. Approximate and asymptotic studies involving slow sources with Markov arrival processes have been reported [24, 25].

There are many potential applications of the queueing system presented in this paper. It may be adopted for the modeling of a multiplexer at the access points of high speed networks receiving traffic from lower speed networks or from sources with a slow packet generation mechanism compared to the network slot. Due to limitations on the information processing speed, the time-constant of most information sources may be larger than that of a high speed (fiber optics) networks. Furthermore, earlier developed (low speed) networks behave like slow information sources at the access point of a high speed backbone network. Similar queueing systems appear in a computer processor or a production line where jobs delivered to a server require a preprocessing time greater than the processing time at the server. Some results on the impact of different network transmission speeds under Markov formulation and by employing the M/G/1 paradigm [23] can be found in Daigle [11].

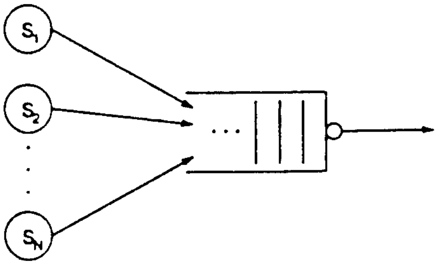


Fig. 1. The statistical multiplexer.

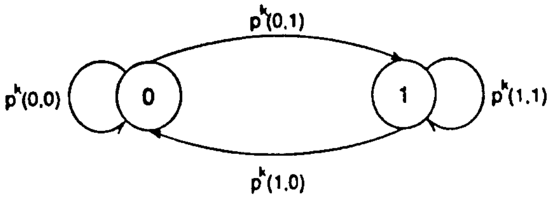


Fig. 2. The first-order Markov model for the packet source.

In the next section, the packet arrival processes generated by some types of slow sources are described, and a Markovian model is constructed for the tractable description of these processes. In Section 3, a general queueing model is formulated and analyzed. This model is employed in the study of the multiplexer under the slow arrival processes. Although the analysis presented here is different, the formulated queueing model can be seen to fall into the general M/G/1 paradigm [23]. Numerical results are presented in the last section, together with a discussion on the impact of the correlation and the source time-constant on the queueing behavior.

2. Source packet traffic and the formulated queueing systems

Consider a fixed-speed slotted network line that supports the transmission of packetized information generated by N sources. The network slot (or network time-constant) is equal to the packet transmission time. Packet transmissions are synchronized with the network slot boundaries. The sequence of the slot boundaries defines the discrete-time axis of the system, denoted by $J; J = \{0, 1, 2 \ldots\}$. The constant length packets generated by the sources are temporarily stored in a buffer of infinite capacity. The formulated queueing system can be modeled in terms of a single server, First-Come First-Served (FCFS), discrete-time queueing system with deterministic and equal to one slot service time, Fig. 1.

Let S^k denote the kth packet source; a superscript k will denote a quantity associated with the kth packet source for $1 \leq k \leq N$. Each of the packet sources is assumed to be either active (state 1) or inactive (state 0). Transitions between the two states occur according to a first-order Markov model (Fig. 2). The one step transition time from state 0 and state 1 may or may not be identical, resulting in single- or double-time-constant sources.

2.1. Single-time-constant sources: asymmetric case

Consider N single-time-constant sources of packetized information. As will be shown below, the packet traffic generated by the kth sources will be described in terms of the parameters $\{\pi^k(1), \gamma^k, \tau^k, c^k\}$ defined as follows, for $1 \leq k \leq N$:

$\pi^k(i)$: It denotes the steady state probability that the k^{th} Markov chain is in state i, $i \in \{0, 1\}$.

γ^k: It denotes the (source) burstiness coefficient defined by $p^k(1,1) - p^k(0,1)$; $p^k(i,j)$ denotes the transition probability that the kth Markov chain moves from state i to state j, $i, j \in \{0, 1\}$.

τ^k: It is defined as the source time-constant and it is equal to the one-step transition time of the k^{th} Markov chain; $\tau^k \in Z^-$, where Z^- is the set of the positive integers. A transition is assumed to occur in the chain every τ^k time units; the same state may be revisited at the end of the transition.

c^k: It is a constant which denotes the time instant when the first transition of the k^{th} Markov chain occurs; $0 \leq c^k \leq \tau^k - 1$, $c^k \in Z^0$, where Z^0 is the set of the non-negative integers.

Note that the steady state and the transition probabilities associated with the k^{th} source can be obtained from $\pi^k(1)$ and γ^k by incorporating the Markov properties. The sequence of time instants T^k, $T^k \subseteq J$, at which transitions of the k^{th} Markov chain occur is given by the set

$$T^k = \left\{ j \in J : \frac{j - c^k}{\tau^k} \in Z^0 \right\}. \tag{1}$$

The packet traffic $\{A_j^k\}_{j \in J}$ delivered by the k^{th} source is a discrete-time process defined in terms of the (underlying) Markov packet generating mechanism. Packets are assumed to be generated at the end of a one-step transition interval of the Markov chain associated with the source, as a result of the source activity during that interval. That is, they may be generated only at some time instant t^k, $t^k \in T^k$. Let i be the state of the k^{th} Markov chain at $t^k - \Delta t$, $t^k \in T^k$, $0 < \Delta t < 1$. The number of packets, n, generated by the source at t^k is described by the probability mass $\phi^k(i,n)$, $i \in \{0, 1\}$, $n \in Z^0$. It is assumed that no packets are generated when the source is inactive. That is, $\phi^k(0,0) = 1$, $1 \leq k \leq N$.

In view of the above discussion $\{A_j^k\}_{j \in J}$ is completely determined in terms of the parameters

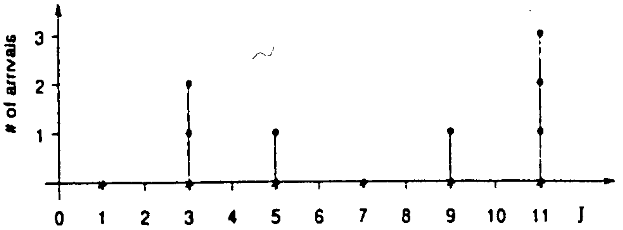


Fig. 3. A realization of the arrival process $\{A_j^k\}_{j \in J}$ for $\tau^k = 2$, $c^k = 1$; * denotes potential arrival points.

$$\{\pi^k(1), \gamma^k, \tau^k, c^k, \phi^k(1,n) \quad \text{for} \quad n \in Z^0\}. \tag{2}$$

Notice that the sequence of time instants of potential packet arrivals from the k^{th} source, T^k, is completely determined from Eq. (2). The same holds true for the set of time instants L^k at which it is guaranteed that no packet will be generated by the k^{th} source. L^k is given by

$$L^k = J - T^k = \left\{ j \in J : \frac{j - c^k}{\tau^k} \notin Z^0 \right\}. \tag{3}$$

Note that if $j \in L^k$ then $j + n\tau^k \in L^k$ as well, for $n \in Z^0$.

The packet arrival process $\{A_j^k\}_{j \in J}$ determined by Eq. (2) is different from those associated with similar queueing systems that have been studied in the past. When $\tau^k = 1$ (which implies that $c^k = 0$), $\{A_j^k\}_{j \in J}$ becomes the standard Markov Modulated Generalized Bernoulli (MMGB) process; here, the term standard implies that the one-step transition time is equal to one time unit. Of course, $\{A_j^k\}_{j \in J}$ for $\tau^k > 1$ may be seen as a standard MMGB process provided that the time unit of the system be equal to τ^k. When packet sources with different time-constants are involved, the selection of the network time-constant for the construction of the system time axis J results in convenient characterization of both the arrival and the departure (from the buffer) processes. One departure occurs at every time instant $j \in J$, provided that the buffer is non-empty. In this paper it is assumed that the time-constant of the sources are integer multiples of the network time-constant. Thus, all arrivals will occur at some time instant in J. A realization of an arrival process $\{A_j^k\}_{j \in J}$ for $\tau^k = 2$ and $c^k = 1$ is shown in Fig. 3.

A packet source that generates an arrival process described by Eq. (2) for some $\tau^k > 1$ is considered to be slow (with respect to the network time constant) and correlated. Such packet sources may be adopted for the modeling of packet arrival processes appearing in communication networks. For instance, the packet traffic delivered by a (slow) transmission line to a multiplexer feeding a network line which is τ^k times faster, can be described in terms of a process determined by Eq. (2). Another example is the traffic delivered to a high speed line by a slow processor. In this case τ^k, $\tau^k > 1$, describes the processing time required before an output (possibly multi-packet) is generated.

It is important to note that the packet arrival process $\{A_j^k\}_{j\in J}$ has a structure which affects significantly the buffer behavior. This process presents a sequence of periodic time instants (contained in L^k) at which no packet is delivered. The existence of these periodic sequences has a positive impact on the intensity of the queueing problems, as it will be illustrated. Furthermore, the larger the value of τ^k, the less significant will be the impact of the correlation in the source on the queueing intensity. When $\tau^k > 1$, the correlation in the arrival process, as seen by the network (server), is said to have been reduced, compared to that present in the corresponding source. A large value of τ^k will allow the network to transmit a large number (up to τ^k) of packets before the next packet(s) is delivered by the correlated source. The special case of the arrival process $\{A_j^k\}_{j\in J}$ which is based on an uncorrelated packet generating mechanism is also interesting and different from an independent and identically distributed packet arrival process. The latter process may deliver packets over consecutive time instants in J; the former may not if $\tau^k > 1$.

In the rest of this section, the cumulative packet arrival process, $\{A_j\}_{j\in J}$ where

$$A_j = \left[\sum_{k=1}^{N} A_j^k\right]. \tag{4}$$

is described in terms of an appropriate MMGB model. Let

$$\Delta_j = \{S^k : j \in T^k\}, \quad j \in J. \tag{5}$$

That is Δ_j is the set of all sources whose underlying Markov mechanism undergoes transition at time instant j. These sources potentially generate packets at that time instant. From the periodicity of the set T^k (see Eq. (3)) turns out that Δ_j is also periodic. Let

$$M = LCM\{\tau^1, \tau^2, \ldots, \tau^N\} \tag{6}$$

where $M = LCM\{\cdots\}$ denotes the least common multiplier of the argument. It is easily shown that

$$\Delta_j = \Delta_{j-nM} \quad \text{for } n \in Z^0 \quad \text{or } \Delta_j = \Delta_{j \bmod M} \quad \text{for } j \in J. \tag{7}$$

For example, for $N = 2$ and $\tau^1 = 2$, $\tau^2 = 3$, $c^1 = 0$, $c^2 = 0$ turns out that $M = 6$ and $\Delta_0 = \{S^1, S^2\}$, $\Delta_1 = \{o\}$, $\Delta_2 = \{S^1\}$, $\Delta_3 = \{S^2\}$, $\Delta_4 = \{S^1\}$, $\Delta_5 = \{o\}$.

Let $\{X_j\}_{j\in J} = \{(I_j^1, I_j^2, \ldots, I_j^N, M_j)\}_{j\in J}$ be an $(N+1)$-dimensional discrete-time process; I_j^k, $I_j^k \in \{0, 1\}$, $1 \le k \le N$, is a random variable describing the state of S^k at time instant j; $\{M_j\}_{j\in J}$ is a periodic Markov chain with state space $\{0, 1, \ldots, M-1\}$ which evolves as described below:

$$M_{j+1} = (M_j + 1) \bmod M \quad \text{with probability } 1. \tag{8}$$

The state of $\{M_j\}_{j\in J}$ determines which of the sources undergo a state transition at the current time. The latter information together with that provided by the process $\{(I_j^1, I_j^2, \ldots I_j^n)\}_{j\in J}$ make $\{X_j\}_{j\in J}$ a Markov process. Let Ω, $\Omega = \{0, 1\}^N \times \{0, 1, \ldots, M-1\}$, denote the state space of $\{X_j\}_{j\in J}$ and let $\bar{\Omega}$, $\bar{\Omega} = 2^N M$, denote its cardinality. The transition probabilities of $\{X_j\}_{j\in J}$ are given by

$$\begin{aligned} p\{(i^1, i^2, \ldots, i^N, n), (\bar{i}^1, \bar{i}^2, \ldots, \bar{i}^N, \bar{n})\} &= \prod_{k: S^k \in \Delta_{(n+1) \bmod M}} p^k(i^k, \bar{i}^k) \prod_{k: S^k \notin \Delta_{(n+1) \bmod M}} 1_{\{\bar{i}^k = i^k\}} \\ &\quad 1_{\{\bar{n} = (n-1) \bmod M\}}, \quad (i^1, i^2, \ldots, i^n, n), \\ &\quad (\bar{i}^1, \bar{i}^2, \ldots, \bar{i}^N, \bar{n}) \in \Omega \end{aligned} \tag{9}$$

where it is defined that

$$\prod_{k: S^k \in \Delta_{n-1 \bmod M}} p^k(i^k, \bar{i}^k) = 1 \text{ if } \Delta_{(n+1) \bmod M} = \{o\}.$$

Notice that only the transition probabilities associated with the Markov chains of the sources in $\Delta_{(n+1) \bmod M}$ are considered, since only these chains undergo a state transition at $j = (n+1) \bmod M$. The first indicator function imposes the condition that the state of the sources not contained in $\Delta_{(n-1) \bmod M}$ remain unchanged. The second indicator function imposes the requirement that the Markov chain $\{M_j\}_{j\in J}$ moves to the next state as determined by Eq. (8).

The cumulative arrival process $\{A^j\}_{j\in J}$ can be determined by the packet generation probabilities

$$o(i^1, i^2, \ldots, i^N, n; m) = \left[\bigotimes_{S^k \in \Delta_{(n+1) \bmod M}} o^k(i^k; \cdot)\right](m), \quad (i^1, i^2, \ldots, i^N, n) \in \Omega, \ m \in Z^0 \tag{10}$$

where

$$\left[\bigotimes_{S^k \in \Delta_{(n+1) \bmod M}} o^k(i^k; \cdot)\right](m) \equiv \begin{cases} 0 \text{ for } m \ne 0 & \text{if } \Delta_{(n-1) \bmod M} = \{o\} \\ 1 \text{ for } m = 0 & \text{if } \Delta_{(n-1) \bmod M} = \{o\} \\ k\text{-fold convolution of } o^k(i^k; \cdot), & \text{for all } k : s^k \in \Delta_{(n-1) \bmod M}, \text{ at } m. \end{cases}$$

Note that packets may be generated only at the transition instants of the corresponding Markov chain and not at every discrete-time instant $j, j \in J$, at which the source is active.

The original queueing system may now be modeled in terms of an equivalent system whose packet arrival process is described by a MMGB process $\{A_j\}_{j \in J}$; $\{A_j\}_{j \in J}$ is described in terms of the Markov chain $\{X_j\}_{j \in J}$ and the packet generating probabilities $\varphi(x;m)$, $x \in \Omega$ given by Eq. (10). This queueing system is studied in the next section.

2.2. *Single-time-constant sources: symmetric case*

When the N packet arrival processes are identical (and synchronized) – that is, when the parameters in Eq. (2) are identical for all processes – then the $(N+1)$-dimensional Markov chain $\{X_j\}_{j \in J}$ can be replaced by the 2-dimensional Markov chain

$$\{X_j^{id}\}_{j \in J} = \{(I_j^{id}, M_j^{id})\}_{j \in J}.$$

$\{M_j^{id}\}_{j \in J}$ is a periodic Markov chain defined as $\{M_j\}_{j \in J}$, with parameter $M = \tau$; I_j^{id} is the random variable which describes the number of active sources at time instant j, $j \in J$. Let Ω^{id}, $\Omega^{id} = \{0, 1, \ldots, N\} \times \{0, 1, \ldots, \tau - 1\}$, denote the state space of $\{X^{id}\}_{j \in J}$; its cardinality $\bar{\Omega}^{id}$ is equal to $(N+1)\tau$. The sets $T^k(L^k)$ defined by Eq. (1) (Eq. (3)) are identical for all k, $1 \le k \le N$, and they are denoted by $T^{id}(L^{id})$. The set of active sources, Δ^{id}, is given by

$$\Delta_j^{id} = \begin{cases} \{\phi\} & \text{if } j \in L \\ \{S^1, S^2, \ldots, S^N\} & \text{if } j \in T. \end{cases} \tag{11}$$

For simplicity it may be assumed that $c^k = 0$ for all k, $1 \le k \le N$. In this case Eq. (11) becomes

$$\Delta_j^{id} = \begin{cases} \{\phi\} & \text{if } j \bmod \tau = 0 \\ \{S^1, S^2, \ldots, S^N\} & \text{otherwise.} \end{cases}$$

The transition probabilities of $\{X_j^{id}\}_{j \in J}$ are given by

$$p^{id}\{(i,n),(\bar{i},\bar{n})\} = [q(i,\bar{i})1_{\{\bar{n}=0\}} + 1_{\{\bar{n} \neq 0, \bar{i}=i\}}] \times 1_{\{\bar{n}=(n+1) \bmod \tau\}}, \tag{12}$$

where the transition probabilities $q(i,\bar{i})$, $0 \le i$, $\bar{i} \le N$, of the Markov chain $\{I_k^{id}\}_{k \in T^{id}}$ – defined on the transition time instants in T^{id} – with unit-step transition time, τ, are given by

$$q(i,\bar{i}) = \sum_{k=\max\{0, i+\bar{i}-N\}}^{\min\{i,\bar{i}\}} \binom{i}{k} p^k(1,1) p^{i-k}(1,0) \times \binom{N-i}{\bar{i}-k} p^{\bar{i}-k}(0,1) p(0,0)^{N-i-(\bar{i}-k)}. \tag{13}$$

The last indicator function in Eq. (12) imposes the condition that the Markov chain $\{M_j\}_{j \in J}$ moves to the next state as determined by Eq. (8). The first indicator function imposes the requirement that a transition occurs at a time instant in T^{id} ($\bar{n} = 0$) in which case the transition probability is given by Eq. (13). The second indicator function imposes the requirement that the state of $\{I_j^{id}\}$ remain unchanged if $(j+1) \notin T^{id}$ ($\bar{n} \neq 0$). If $\bar{n} \neq 0$ and $\bar{i} = i$, the transition probability is equal to one, provided that $\bar{n} = (n+1) \bmod M$.

The cumulative packet arrival process $\{A_j^{id}\}_{j \in J}$ can be described in terms of a MMGB process based on the underlying Markov chain $\{X_j^{id}\}_{j \in J}$ and the packet generating probabilities given by

$$\varphi^{id}(i,n;m) = \begin{cases} 0 \text{ for } m \neq 0 & \text{if } (n+1) \bmod M \neq 0 \text{ or } i = 0 \\ 1 \text{ for } m = 0 & \text{if } (n+1) \bmod M \neq 0 \text{ or } i = 0 \\ [\circledast_i \varphi^{id}(1:\cdot)](m) & \text{if } n+1 = 0, \end{cases} \tag{14}$$

where $\circledast_i$ denotes an i-fold convolution and $\varphi^{id}(1;\cdot)$ denotes the common packet generating probability associated with a single source, as employed in Eq. (2).

Although the cumulative packet arrival process in the case of identical packet sources may be described in terms of $\{X_j\}_{j \in J}$ (and the corresponding probabilities be given by Eq. (10)), the formulation of $\{X_j^{id}\}_{j \in J}$ results in a state space Ω^{id} whose cardinality $\bar{\Omega}^{id}$ increases linearly with respect to N; on the other hand, $\bar{\Omega}$ increases exponentially with respect to N. As it will be seen in the next section, the cardinality at the underlying Markov chain affects significantly the complexity of the solution.

2.3. *Double-time-constant sources*

In the packet arrival processes considered so far, it has been assumed that the underlying Markov mechanism of the source makes a transition every τ^k time units. As a result, the set T^k determines the sequence of time instants in J at which activation of the source may occur. In some applications it is reasonable to assume that it takes τ^k time units for the source to generate (prepare) a packet, when active. On the other hand, the re-activation of the source may occur at any time instant and not necessarily at some integer multiple of τ^k. A correlated source with this behavior may be described in terms of the Markov model shown in Fig. 2, with two different one-step transition times (double-time-constant source). A transition from state 1 (active) occurs every τ^k time units while a transition from state 0 (inactive) occurs every one time unit. The packet arrival process generated by such a source is described by the parameters in Eq. (2) with the understanding that transition from a state 0 may occur in the next slot. Under this modeling, the re-activation of the source may occur at any time instant in J and not necessarily at the time instants contained in T_k. The length of an active period is geometrically

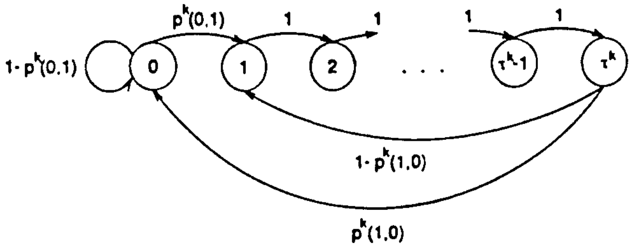


Fig. 4. The Markov model for $\{Y_j^k\}_{j\in J}$.

distributed with mean $1/p^k(1,0)$ in time units of length τ^k; the length of an inactive period is geometrically distributed with mean $1/p^k(0,1)$ in system time units.

The double-time-constant Markov source introduced above may be alternatively described in terms of the Markov chain $\{Y_j^k\}_{j\in J}$ shown in Fig. 4. Let Ω_y^k, $\Omega_y^k=\{0,1,\ldots,\tau^k\}$ denote its state space whose cardinality $\bar{\Omega}_y^k$ is equal to τ^k+1. Let $p_y^k(i,j), i,j\in\Omega_y^k$, denote the transition probabilities of $\{Y_j^k\}_{j\in J}$; the non-zero such probabilities are given by

$$p_y^k(i,j)=1,\ j=i+1, 1\le i<\tau^k,$$
$$p_y^k(\tau^k,1)=1-p^k(1,0). \qquad (15)$$
$$p_y^k(\tau^k,0)=p^k(1,0), \quad p_y^k(0,1)=p^k(0,1),$$
$$p_y^k(0,0)=1-p^k(0,1). \qquad (15)$$

Note that state 0 in Fig. 4 corresponds to the inactive state 0 (Fig. 2). A visit to state τ^k determines a time instant at which the active source will undergo a state transition. The source moves from state τ^k to state 0 (becomes inactive) in the next time instant with probability $p_y^k(\tau^k,0)=p^k(1,0)$; the source moves from state 0 to state 1 with probability $p_y^k(0,1)=p^k(0,1)$. The source remains in the active state for τ^k time units. Upon visit to state τ^k packets are generated according to the probabilities

$$\phi_y^k(\tau^k;m)=\phi^k(i;m),\ m\in Z^0. \qquad (16)$$

No packets are generated from the other states. That is,

$$\phi_y^k(i;0)=1,\ 0\le i<\tau^k. \qquad (16)$$

In view of the above discussion, the packet arrival process $\{B_j^k\}_{j\in J}$, generated by the kth double-time-constant source, $1\le k\le N$, can be described as a MMGB process based on the underlying Markov chain $\{Y_j^k\}_{j\in J}$ and the packet generating probabilities given by Eq. (16).

The cumulative packet arrival process $\{A_j^r\}_{j\in J}$, generated by the arrival processes $\{B_j^k\}_{j\in J}$, $1\le k\le N$, can be described as a MMGB process based on the N-dimensional Markov chain $\{X_j^r\}_{j\in J}=\{(Y_j^1,\ldots Y_j^N)\}_{j\in J}$ with state space $\Omega^r=\Omega_y^1\times\Omega_y^2\times\Omega_y^N$, transition probabilities

$$p^r((i^1,i^2,\ldots,i^N),(\bar{i}^1,\bar{i}^2,\ldots,\bar{i}^N))=\prod_{k=1}^{N}p_y^k(i^i,\bar{i}^k) \quad (17)$$

and packet generating probabilities

$$\phi^r(y^1,y^2,\ldots y^N;k)=[\underset{k}{\circledS}\phi_y^k(y^k;\cdot)](k),\ k\in Z^0,$$
$$(y^1,y^2,\ldots y^N)\in\Omega_r. \qquad (18)$$

In the next section the analysis of the queueing model formulated above is presented.

3. The queueing system under MMGB arrivals

3.1. The general case

Consider the FCFS, single server, discrete-time queueing systems with cumulative packet arrivals described by a MMGB process $\{A_j\}_{j\in J}$; J denotes the system axis, as defined in the previous section. Let $\{X_j\}_{j\in J}$ denote the underlying Markov process associated with $\{A_j\}_{j\in J}$; let Ω denote its state space with cardinality $\bar{\Omega}$; let $\pi(x)$ and $p(x,y)$ denote its steady state transition probabilities, $x,y\in\Omega$. Let $\phi(x;k)$ denote the probabilistic mapping which determines the probability that k packets arrive when $\{X_j\}_{j\in J}$ is in state $x, x\in\Omega$, $k\in Z^0$ and $k\le R$, $0<R\le\infty$; let $\mu(x)$ and $\sigma(x)$ denote the first and the second moments, respectively, of the probability mass function $\phi(x;\cdot)$.

Let $p^m(j;y)$ denote the joint probability that there are j packets in the (queueing) system at the m^{th} time instant (packet arrivals at this time instant are included) and the state of $\{X_j\}_{j\in J}$ is y; the packet arrivals generated due to the visit to state y have not been accounted for. The evolution of the buffer occupancy process can be described by an $(\bar{\Omega}+1)$-dimensional Markov chain with state space $\bar{\Omega}\times\{0,1,2,\ldots\}$ and state probabilities described by the equations:

$$p^m(j;y)=\sum_{x\in\Omega}\sum_{k=0}^{R}p^{m-1}(j+1-k;x)p(x,y)\phi(x;k),$$
$$j\ge R+1 \qquad (19a)$$

$$p^m(j;y)=\sum_{x\in\Omega}p(x,y)\sum_{k=1}^{j-1}[p^{m-1}(k;x)\phi(x;j+1-k)$$
$$+p^{m-1}(0;x)\phi(x;j)],\quad 0\le j\le R. \qquad (19b)$$

Under the stability condition for the queueing system, given by

$$\lambda=\sum_{x\in\Omega}\mu(x)\pi(x)<1 \qquad (20)$$

the $(\bar{\Omega}+1)$-dimensional Markov chain described in

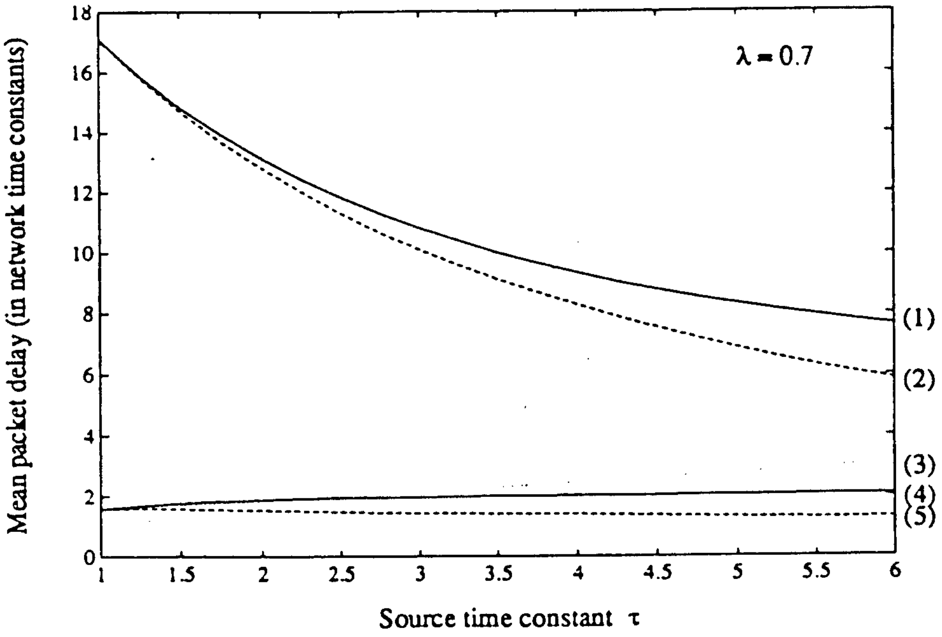


Fig. 5. Mean delay results for the packetized voice sources.

Eq. (19) is ergodic. By considering the limiting probabilities as $m \rightarrow \infty$ and their generating function, the following equations are derived:

$$P(z;y) = \sum_{k=0}^{R} \sum_{x \in \Omega} z^{k-1}[P(z;x) + (z-1)p(0;x)] \times p(x,y)\phi(x;k) \qquad (21)$$

$p(0;x)$ denotes the boundary probability that the system is empty and the state of the packet arrival process is x; its computation is discussed in Appendix A. $P(z;y)$ denotes the probability generating function of $p(j;y)$, $0 \le j \le \infty$. Note that

$$P(z) = \sum_{y \in \Omega} P(z;y) \qquad (22)$$

where $P(z)$ denotes the probability generating function of the buffer occupancy process. Since the k^{th} derivative of $P(z)$, evaluated at $z = 1$, is equal to the k^{th} factorial moment of the number of packets in the system [21], it turns out that the k^{th} moment can be evaluated by differentiating Eq. (21) k times, setting $z = 1$ and solving the resulting system of equations. These equations are not linearly independent. An additional, linearly independent equation is derived by following a procedure similar to that shown in Appendix B (for the first moment) and invoking L' Hospital's rule. By differentiating Eq. (21) and setting $z = 1$ the following system of $\bar{\Omega}$ linear equations is obtained with respect to $P'(1;x)$, $x \in \Omega$.

$$P'(1;y) = \sum_{x \in \Omega} p(x,y)[P'(1;x) + (\mu(x) - 1)\pi(x) + p(0;x)], \quad y \in \Omega \qquad (23a)$$

The linearly independent equation (Appendix A) is given by

$$\sum_{x \in \Omega} [2(\mu(x) - 1)P'(1;x) + 2(\mu(x) - 1)p(0;x) + (2 + \sigma(x) - 3\mu(x))\pi(x)] = 0 \qquad (23b)$$

The mean buffer occupancy, $Q_1 = P'(1)$, and the mean packet delay, D (by using Little's theorem), are then obtained from

$$Q_1 = \bar{I}\bar{P}_1^T, \quad D = \frac{Q_1}{\lambda} \qquad (24)$$

where $\bar{I}$ is the $\bar{\Omega}$-dimensional identity row vector, $\bar{P}_1^T$ is the transpose of the row vector of $P'(1;x)$, $x \in \Omega$, and λ is the mean packet arrival rate given by Eq. (20).

If $\{A_j\}_{j \in J}$ is an independent and identically distributed process, it may be considered to be a MMGB process based on an underlying Markov chain with a single state, x_0. In this case $\Omega = \{x_0\}$, $\pi(x_0) = 1$, $p(x_0,x_0) = 1$, $\mu(x_0) = \lambda$ and $\sigma(x_0) = \sigma$. From Eqs. (23), (24) it is easily obtained that

$$Q_1^{ind} = \lambda + \frac{\sigma - \lambda}{2(1-\lambda)} \quad \text{and} \quad D^{ind} = 1 + \frac{\sigma/\lambda - 1}{2(1-\lambda)}. \qquad (25)$$

Eqs. (23) and (24) provide for the mean packet delay induced by the queueing system under all source models considered in Section 2. The dimensionality of this system of equations is equal to the cardinality of the underlying Markov chain $\{X_j\}_{j \in J}$. Under single-time-constant sources and asymmetry (Section 2.1), the underlying Markov chain $\{X_j\}_{j \in J}$ has a state space of cardinality $2^N M$; its transition probabilities are given by Eq. (9) and the associated packet generating probabilities by Eq. (10). Under single-time-constant sources and symmetry (Section 2.2), the underlying Markov chain $\{X_j^{ud}\}_{j \in J}$ has a state space of cardinality $(N+1)\tau$; its transition probabilities are given by Eqs. (12), (13) and the associated packet generating probabilities by Eq. (14). Notice that the number of equations which need to be solved increases exponentially for the asymmetric case and linearly for the symmetric case, with respect to the number of sources N. Under double-time-constant sources (Section 2.3), the underlying Markov chain $\{X_j^r\}$ has a state space of cardinality $\prod_{k=1}^{n}(\tau^k+1)$; its transition probabilities are given by Eq. (17) and the associated packet generating probabilities by Eq. (18). When the double-time-constant sources are identical with time-constant τ, the number of linear equations which need to be solved can be significantly smaller than $K_{\mathrm{asym}} = (\tau+1)^N$. In Appendix C it is shown that the number of equations in this case is given by

$$K_{\mathrm{sym}} = \frac{(N-\tau)!}{N!\tau!}. \tag{26}$$

The reduced set of linear equations and a discussion on the numerical complexity may be found in Appendix C.

4. Results and conclusions

In this section, numerical results are presented for the mean packet delay induced by a statistical multiplexer under the packet arrival processes described in Section 2.

In Fig. 5 some numerical results are presented, as a function of the source time-constant τ, under various packet traffic models. The packet sources are assumed to be identical with parameters

$$\{\pi^k(1) = 0.35, \gamma^k = 0.93, \tau, c^k = 0, \rho^k(1,1) = 1\},$$
$$1 \le k \le N,$$

where the number of sources, N, is set to be equal to 2τ. The resulting cumulative packet traffic rate is equal to $N \pi^k(1)/\tau = 0.70$ packets time units. Notice these packet sources have the characteristics of packetized voice sources delivering one packet every τ time units, when active.

Curve (1) in Fig. 5 presents the induced packet delay. Notice that for the same total traffic rate, the induced packet delay decreases as the time-constant increases. This decrement in the delay is attributed to the reduction of the source correlation ($\gamma^k = 0.93$) as seen by the server (as τ increases), as well as to the increased independence in the cumulative traffic due to the increased number of (mutually independent) contributing sources ($N = 2\tau$). The cumulative arrival process $\{A_j^{ud}\}_{j \in J}$ is determined as described in Section 2.2. In this case, $(2\tau+1)\tau$ linear equations are contained in Eq. (23).

Curve (2) presents the induced packet delay when c^k, $1 \le k \le N$, are not identical for all sources but they are spread nearly uniformly over τ. When $\tau = 1$, no such spreading is possible and $c^k = 0$, $1 \le k \le N$. In this case curves (1) and (2) coincide. Notice that the larger the value of τ, the larger the smoothing of the arrival process that can be accomplished through the spreading of c^k. The intensity of the queueing problems is decreased as τ increases, as a result of the distribution of the potential packet arrival instants associated with the various sources. The cumulative arrival process $\{A_j\}_{j \in J}$ is determined as described in Section 2.1. In this case, $2^{2\tau}\tau$ linear equations are contained in Eq. (23).

Curve (3) presents the mean packet delay results when the packets are delivered by each of the $N = 2\tau$ sources according to a Bernoulli process with rate $\pi^k(1)/\tau$ packets per time unit. The delay results, in this case, are obtained from Eq. (25). The resulting cumulative packet arrival process is a binomial with parameters 2τ and $\pi^k(1)/\tau$. The second moment of this process is given by

$$\sigma = N\frac{\pi^k(1)}{\tau}\left[1-\frac{\pi^k(1)}{\tau}\right] + \left[N\frac{\pi^k(1)}{\tau}\right]^2$$
$$= 2\pi^k(1)\left[1-\frac{\pi^k(1)}{\tau}\right] + [2\pi^k(1)]^2.$$

Notice that σ increases as τ increases, and thus the mean packet delay also increases, according to Eq. (25). This trend is clearly observed in curve (3) of Fig. 5. Notice that as τ increases the difference between curves (1) (or (2)) and curve (3) decreases, as expected.

Curve (4) presents the delay results under the traffic modeling considered for curve (1) but assuming that $\gamma^k = 0$, $1 \le k \le N$. That is, the sources are assumed to be uncorrelated. Notice that the resulting cumulative packet arrival process is different from the binomial process which was considered for the derivation of curve (3). For $\gamma^k = 0$, $1 \le k \le N$, and $\tau > 1$, the source can generate packets only at time instants in T^k (separated by τ time units) and not at any time instant in J, which is the case under the underlying Bernoulli model assumed for curve (3). Notice that the delay results have smaller values in this case compared to those under curve (3), possibly due to the positive effect of the constraint that packets cannot be generated at time instants separated by less than τ time units under the model for curve (4). The cumulative arrival process $\{A_j^{ud}\}_{j \in J}$ is determined as described in Section 2.2. In this case, τ linear equations are contained in Eq. (23), since the underlying Markov chain $\{X_j^{ud}\}_{j \in J}$ becomes one dimensional and it is described by $\{M_j^{ud}\}_{j \in J}$.

Finally, curve (5) presents the delay results under the model considered for curve (4), but under the assumption that the values of c^k, $1 \le k \le N$, are near-uniformly

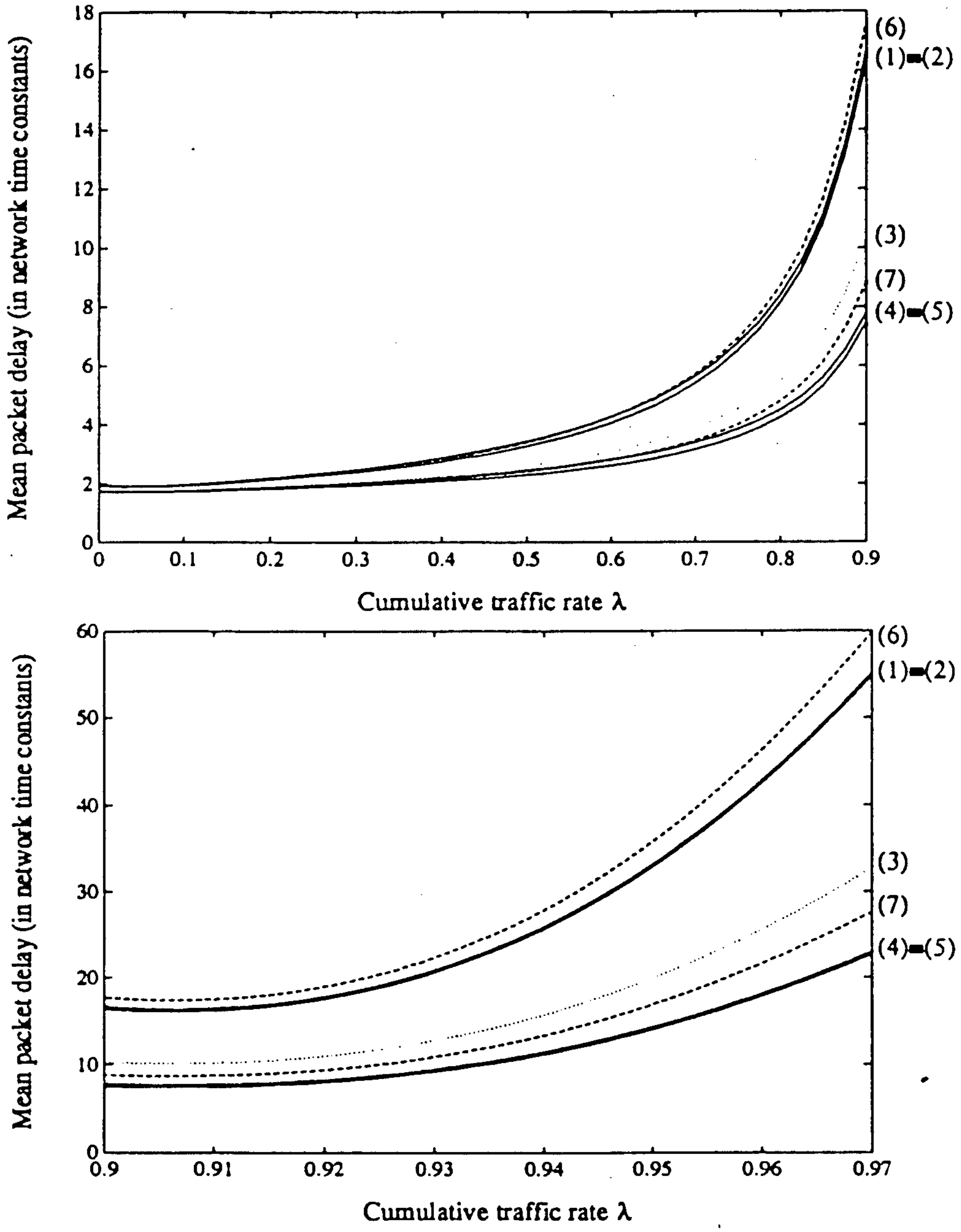


Fig. 6. (a) Mean delay results for the $N = 3$ packet sources under various traffic models; $0 \leq \lambda \leq 0.9$; (b) mean delay results for the $N = 3$ packet sources under various traffic models; $0.9 \leq \lambda \leq 0.97$.

spread over τ. As it was explained before, this spreading of the potential packet arrival instants has a positive impact on the queueing intensity; this effect is clearly observed by comparing curves (4) and (5) of Fig. 5. The cumulative arrival process $\{A_j\}_{j \in J}$ is determined as described in Section 2.2. In this case, $M = LCM\{\tau, \tau, \ldots, \tau\} = \tau$ linear equations are contained in Eq. (23), since the underlying Markov chain $\{X_j\}_{j \in J}$ becomes one dimensional and it is described by $\{M_j\}_{j \in J}$.

Notice that when $\tau = 1$ the arrival processes may change state at any slot. In this case, the results may also be obtained from the closed form expression given by [4]:

$$\hat{D} = \left[1 + \frac{\sum_{n=1}^{N}\sum_{m>n}^{N} \lambda^n \lambda^m \left(1 + \frac{\gamma^n}{1-\gamma^n} + \frac{\gamma^m}{1-\gamma^m}\right)}{\left(1 - \sum_{n=1}^{N} \lambda^n\right) \sum_{n=1}^{N} \lambda^n}\right]. \quad (26)$$

Fig. 6 presents the delay results obtained by multiplexing $N = 3$ packet sources with different time-constants given by $\tau^1 = 2$, $\tau^2 = 4$ and $\tau^3 = 4$. The rest of the parameters are assumed to be identical. The source of burstiness coefficient is equal to $\gamma^{id} = 0.5$ and the packet generating probabilities are given by $\phi^{id}(1,1) = 0.3$, $\phi^{id}(1,2) = 0.4$ and $\phi^{id}(1,3) = 0.3$. The probability that the packet source is in the active state, $\pi^{id}(1)$, is selected so that the resulting cumulative packet traffic rate is equal to some value from the horizontal axis of Fig. 6. Curves (1), (2), (3), (4) and (5) show the mean packet delay under the traffic modeling considered for the derivation of the corresponding curves of Fig. 5. Curve (6) is derived under the assumption that each of the sources may be re-activated at any time instant and not at integer multiples of τ^k, $1 \leq k \leq 3$. Thus, the double-time-constant source model, discussed in Section 2.3, has been assumed. $\prod_{k=1}^{N}(\tau^k + 1)$ linear equations are contained in Eq. (23), in this case. By exploiting the partial symmetry ($S^2 = S^3$) and properly formulating the equations in Eq. (23) (see Appendix C) a reduced number of linear equations will need to be solved. Curve (7) presents results under the source modeling considered for curve (6) under the assumption that $\gamma^k = 0$, $1 \leq k \leq 3$. Notice that states 0 and 1 may be merged in this case (Fig. 4), resulting in a reduced number of $\prod_{k=1}^{N} \tau^k$ linear equations contained in Eq. (23). Further reduction may be achieved by exploiting the partial symmetry in the arrival process, as explained before.

Notice that the coincidence of curves (1) and (2) and curves (4) and (5) is due to the specific parameters of the selected example. Finally, notice that curve (6) is always above curves (1) and (2) derived for the same values of γ^k, $1 \leq k \leq 3$. Similarly, curve (7) is always above curves (4) and (5) derived for $\gamma^k = 0$, $1 \leq k \leq 3$. This behavior may be explained in view of the fact that the minimum separation between two consecutive active periods in the traffic for the curves (1), (2), (4) and (5) is equal to τ^k, $1 \leq k \leq 3$, while that under the traffic models for the curves (6) and (7) is equal to one. The latter is expected to have negative effect on the intensity of the resulting queueing problems. Finally, the increased delay results presented under curve (3) may be explained in view of the fact that there is no minimum separation between consecutive packet arrivals, in any case, for the traffic model for curve (3).

Acknowledgements

Research supported in part by the Advanced Research Project Agency (ARPA) under Grant F49620-93-1-0564 monitored by the Air Force Office of Scientific Research (AFOSR) and the National Science Foundation under Grant NCR-9011962.

Appendix A

The boundary probabilities $p(0;x)$ are obtained as follows. When there is at most one stage x_0 in Ω such that $\phi(x_0;0) > 0$ (that is, x_0 is the only state which may not deliver a packet to the system), then it is easily established that

$$p(0;x) = q_0 p(x_0, x), \tag{A.1}$$

where $q_0 = 1 - \lambda$ is the probability that the system is empty. Notice that x_0 is the only state at which $\{X_j\}_{j \epsilon J}$ could have been in the previous slot, so that the system is currently empty. When the previous condition is not satisfied, the boundary probabilities may be obtained through the following procedure which may be found in Ref. [10]:

$$p(0;x) = p(0;i) = q_0 \Pi_w(i) \quad \text{for } x \epsilon \Omega \text{ or } i \epsilon \{1, 2, \ldots, \Omega\}, \tag{A.2}$$

where i, $1 \leq i \leq \bar{\Omega}$, denotes an ordering of the states in Ω; $\pi_w(i)$, $1 \leq i \leq \bar{\Omega}$, are computed from

$$\bar{\Pi}_w \bar{e} = 1, \quad \bar{\Pi}_w [P_w] = \bar{\Pi}_w \tag{A.3}$$

where $\bar{\Pi}_w = [\pi_w(1), \ldots, \pi_w(\bar{\Omega})]$ and the matrix $[P_w]$ is computed from

$$[P_w] = \sum_{k=0}^{R} [G_k][P][P_w]^k, \tag{A.4}$$

where $[P]$ is the transition probability matrix of the Markov chain $\{X_j\}_{j \epsilon J}$ and $[G_k]$ is an $\bar{\Omega} \times \bar{\Omega}$ diagonal matrix, $[G_k] = \text{diag}[\phi(1;k), \phi(2;k), \ldots, \phi(\bar{\Omega};k)]$, $0 \leq k \leq R$; R is the maximum number of packet arrivals over a slot; $\{\phi(i;k)\}_{i=1}^{\Omega} \equiv \{\phi(x;k)\}_{x \epsilon \Omega}$, $0 \leq k \leq R$.

Appendix B

The linearly independent equation shown at Eq. (23b) is derived by differentiating Eq. (21); adding up the resulting equations and using the derivative of Eq. (22):

$$\begin{aligned} P'(z) = \sum_{k=0}^{R} \sum_{x \epsilon \Omega} \phi(x;k) \Big[& (k-1) z^{k-2} \\ & \times [P(z;x) + (z-1) p(0,x)] + z^{k-1} \\ & \times [P'(z;x) + p(0;x)] \Big] \end{aligned}$$

By adding

$$ZP'(z) - z \sum_{k=0}^{R} \sum_{x \epsilon \Omega} P'(z;x) \phi(x;k) = 0$$

to the right-hand side of the previous equation, the following expression is obtained

$$P'(z) = \frac{A(z)}{1-z} \tag{B.1}$$

where

$$A(z) = \sum_{k=0}^{R} \sum_{x \in \Omega} \phi(x;k)\Big[(k-1)z^{k-2}[P(z;x) \\ +(z-1)p(0;x)] + z^{k-1}[P'(z;x) + p(0;x)] \\ -zP'(z;x)\Big].$$

Since $P'(1)$ is equal to the mean buffer occupancy, which is finite if Eq. (20) holds, and since $1 - z = 0$ for $z = 1$, $P'(1)$ is computed from Eq. (B.1) by using L' Hospital's rule. Thus,

$$P'(z)\Big|_{z=1} = \frac{dA(z)/dz}{d(1-z)/dz}\Big|_{z=1} = -dA(z)/dz\Big|_{z=1}. \qquad \text{(B2)}$$

From Eq. (B.2) and the derivative of Eq. (22) evaluated at $z = 1$, Eq. (23b) is easily derived.

Table 1
Numerical complexity for the symmetric (K_{sym}) and the asymmetric (K_{asym}) cases as a function of N and $\tau + 1$ (* indicates a number greater than 9,999,999)

N	$\tau + 1$	K_{sym}	K_{asym}
2	31	496	961
	44	990	1,936
3	10	220	1,000
	17	816	4,913
5	4	56	1,024
	8	792	32,768
10	2	11	1,024
	3	66	59,049
	4	286	1,048,576
	5	1,001	9,765,625
15	2	16	32,768
	3	136	*
	4	816	*
20	2	21	1,048,576
	3	213	*
30	2	31	*
	3	496	*
43	2	44	*
...	...	...	...
N	2	$N + 1$	2^N

Appendix C

Let $\tau + 1$ be the cardinality of the state space of the (assumed identical) underlying Markov chains $\{Y_j^k\}_{j \in J}$ (see Section 2.3). Let $\bar{x} = (x_1, x_2, \ldots, x_{\tau+1}) \epsilon [\Omega_r^k]^k$ denote a state of the underlying N dimensional Markov chain $\{X_j^r\}_{j \in J}$. Since the packet arrival processes $\{B_j^k\}_{j \in J}$ are identical, the resulting cumulative arrival process $\{A_j^r\}_{j \in J}$ will assume (probabilistically) the same values under states $\bar{x} \in \Omega_r$ which correspond to the same number of sources in each state; for instance states $(i_1, i_2, \ldots, i_{\tau+1})$ and $(i_2, i_1, \ldots, i_{\tau+1})$.

If $\bar{v}(\bar{x}) = (v_1(x), v_2(\bar{x}), \ldots v_{\tau+1}(\bar{x}))$ is a $(\tau + 1)$-dimensional vector with $v_i(\bar{x})$, $i = 1, 2, \ldots \tau + 1$, denoting the number of input Markov chains at state i, then each such vector $\bar{v}(\bar{x})$ with the constraint $\sum_{i=1}^{\tau+1} v_i(\bar{x}) = N$, represents a class of equivalent states $\bar{x}$. The number of equivalent states $\bar{x}$ in a class $\bar{v}(\bar{x})$ is given by the multinomial coefficient [22]

$$c(\bar{x}) = \begin{pmatrix} N \\ v_1(\bar{x}), v_2(\bar{x}), \ldots v_{\tau+1}(\bar{x}) \end{pmatrix} \\ = \frac{N!}{v_1(\bar{x})! v_1(\bar{x})! \ldots v_{\tau+1}(\bar{x})!}, \qquad \text{(C.1)}$$

where $v_i(\bar{x})$ is the number of input processes in state i, as determined by $\bar{x}$, $i = 1, 2, \ldots, \tau + 1$. This is basically the number of ways in which a population of N elements can be divided into $\tau + 1$ ordered parts of which the first contains $v_1(\bar{x})$ elements, the second contains $v_2(\bar{x})$ elements and so forth. Let F be the set of representative states $\bar{x}$ of the symmetric system (i.e. no two states $\bar{x} \epsilon F$ belong to the same class of equivalent states). Let $v(\bar{x}_0)$ be the class of the equivalent to $\bar{x}_0$ states. For each $\bar{x}_0$, $\bar{y}_0 \epsilon F$, Eqs. (23a) and (23b) can be written as follows:

$$P'(1;\bar{y}_0) = \sum_{\bar{x}_0 \epsilon F} \left\{ \sum_{\bar{x} \epsilon v(\bar{x}_0)} p(\bar{x}, \bar{y}_0) \right\} P'(1;\bar{x}) \\ + \sum_{\bar{x} \epsilon \Omega'} p(\bar{x}, \bar{y}_0) [(\mu(\bar{x}) - 1)\pi(\bar{x}) + p(0;\bar{x})], \\ \bar{y}_0 \epsilon F \qquad \text{(C.2a)}$$

$$\sum_{x_0 \epsilon F} c(\bar{x}_0) 2(\mu(\bar{x}_0) - 1) P'(1;\bar{x}_0) \\ + \sum_{x \epsilon \Omega'} \{2(\mu(\bar{x}) - 1) p(0;\bar{x}) \\ + [2 + \sigma - 3\mu(\bar{x})]\pi(\bar{x})\} = 0, \qquad \text{(C.2b)}$$

where $c(x_0)$ is given by (C.1). The mean buffer occupancy can be obtained from

$$Q_1 = P'(1) = \sum_{x_0 \epsilon F} P'(1;\bar{x}_0) c(\bar{x}_0). \qquad \text{(C.3)}$$

The induced average packet delay is then obtained from Eq. (24). The reduced number of equations in Eq. (C.2) are then easily seen to be given by Eq. (26), which expresses the number of ways of partitioning N things into $\tau + 1$ groups [22]. Notice that significant reduction of the number of required equations may be achieved under symmetry. Comparison of k_{sym} (see Eq. (26)) and $k_{asym} = (\tau + 1)^N$ for some pairs (N, τ) which result in less than 1000 linear equations in Eq. (C.2), are shown in Table 1.

References

[1] H. Heffes and D. Lucantoni, A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer analysis, IEEE J. Select. Areas Commun., 6(4) (September 1986).

[2] D. Towsley, The analysis of a statistical multiplexer with nonindependent arrivals and errors, IEEE Trans. Comm., 1(28) (January 1980).

[3] H. Bruneel, Queueing behavior of statistical multiplexers with correlated inputs. IEEE Trans. Comm., 12(36) (December 1988).

[4] A. Viterbi, Approximate analysis of time synchronous packet networks, IEEE J. Select. Areas Comm.,, 6(4) (September 1986).

[5] M. Reiser, Performance evaluation of data communication systems, Proc. IEEE, 70 (February 1982).

[6] M.S Ali Khan, Infinite dams with inputs forming a Markov chain. J. Appl. Probab., 5 (1968) 72-84.

[7] M.S. Ali Khan, Finite dams with inputs forming a Markov chain. J. Appl. Probab., 7 (1970) 291-303.

[8] J. Daigle, Y. Lee and N. Magalhaes, Discrete time queues with phase dependent arrivals, IEEE INFOCOM Conf., San Francisco, June 5-7, 1990.

[9] I. Stavrakakis, Analysis of a statistical multiplexer under a general input traffic model, IEEE INFOCOM Conf., San Francisco, June 5-7, 1990.

[10] I. Stavrakakis, Efficient modeling of merging and splitting processes in large networking structures, IEEE J. Select. Areas Comm., 9(8) (October 1991).

[11] J. Daigle, Queueing Theory for Telecommunications, Addison Wesley, 1992.

[12] D. Anick, D. Mitra and M. Sondhi, Stochastic theory of data-handling system with multiple sources, Bell Syst. Tech. J., 61(8) (October 1982).

[13] M. Rubinovitch, The output of a buffered data communication system, Stoch. Proces. and their applic., 1 (1973) 375-382.

[14] J. Cohen, Superimposed renewal processes and storage with gradual input, Stoch. Proces. and their Applic., 2 (1974) 31-58.

[15] H. Kaspi and M. Rubinovitch, The stochastic behavior of a buffer with non-identical input lines, Stoch. Proces. and their Applic., 23 (1975) 73-88.

[16] L. Kosten, Stochastic theory of a multi-entry buffer (I), Delft Progress Rep., Math. Engin., Math. and Inf. Engin., 1 (1974) 10-18.

[17] L. Kosten, Stochastic theory of a multi-entry buffer (II), Delft Progress Rep., Math. Engin., Math. and Inf. Engin., 1 (1974) 44-50.

[18] T. Stern, Analysis of separable Markov-modulated queueing processes, Tech. Rep. of CTR, No. CU CTR TR -113-88-41, Columbia University, 1988.

[19] B. Gopinath and J. Morrison, Discrete-time single server queues with correlated inputs. Bell Syst. Tech. J., 56(9) (November 1977).

[20] A. Fraser, B. Gopinath and J. Morrison, Buffering of slow terminals, Bell Syst. Tech. J., 57(8) (October 1978).

[21] D. Heyman and M. Sobel, Stochastic Models in Operations Research, McGraw-Hill, 1982.

[22] H. Stark and J. Woods, Probability, Random Processes and Estimation Theory for Engineers, Prentice Hall, 1986.

[23] M.F. Neuts, Structured Stochastic Matrices of M G 1 Type and their Applications, Marcel Dekker, New York, 1989.

[24] S.Q. Li and H. Sheng, Discrete queueing analysis of multi-media traffic with diversity of correlation and burstiness properties, IEEE INFOCOM, Miami, FL, April 1991.

[25] K. Sohraby, On the asymptotic behavior of heterogeneous statistical multiplexer with applications, IEEE INFOCOM, Florence, Italy, May 1992

ELSEVIER

Performance Evaluation 27&28 (1996) 347–365

PERFORMANCE EVALUATION
An International Journal

Achievable QoS and scheduling policies for integrated services wireless networks [*]

Jeffrey M. Capone [*], Ioannis Stavrakakis

Electrical & Computer Engineering, Northeastern University, Boston, MA 02115, USA

Abstract

Networks are expected to support applications with diverse traffic characteristics and Quality of Service (QoS) requirements, such as voice, data, video and multi-media. In this environment it is important to determine whether a given QoS vector is achievable (for call admission control), and if it is, design efficient resource allocation policies that deliver it.

In this paper a system of T resources (wireless channels) shared by a number of un-buffered applications is considered. Packets which are not allocated a channel within the current assignment cycle are dropped. The QoS vector is equivalently described in terms of (diverse) packet dropping probabilities (p_i), or (diverse) packet dropping rates (d_i). The region of achievable QoS vectors is precisely determined and easy to implement scheduling policies that deliver the achievable QoS vectors are derived. Some applications to call admission control and resource management are presented along with some numerical examples.

Keywords: Integrated services; Wireless networks; Achievable QoS; Scheduling

1. Introduction

The problem of allocating shared resources is an interesting and continuously revisited one. The primary resources to be shared in a networking environment are memory and transmission bandwidth. As technology advances and resources increase their capabilities, existing services are enhanced and new services are provided, driving the need to develop resource allocation technologies that integrate diverse services.

Networking resources are typically managed at two largely different time scales, determined by the service request inter-arrival and duration time scale and a smaller time scale over which certain amount of resources need to be allocated. Management of resources at the larger time scale is the responsibility

[*] Research supported in part by the Advanced Research Project Agency under Grants F49620-93-1-0564 monitored by the Air Force Office of Scientific Research (AFOSR) and MDA972-93-1-0023 monitored by the Army Research Office (ARO) and a Grant from GTE Corporation.

[*] Corresponding author. E-mail: capone@cdsp.neu.edu.

PII S0166-5316(96)00018-1

of the call admission controller. Management of resources at the smaller time scale is the task of the buffer/bandwidth managers.

Typically, both call admission and buffer/bandwidth management functions need to be employed to ensure (sometimes in a probabilistic sense) the delivery of a target Quality of Service (QoS). The more stringent and diverse the target QoS of the supported applications, the more critical is the exercise of the aforementioned management and the more sophisticated their functions become. Typically, the more stringent the target QoS, the greater the importance and need for admission management. This is the case in the telephone network supporting voice services with stringent QoS requirements. In traditional computer data networks where the QoS is not considered to be stringent, the role of call admission management is minimal (if any). While "uniform" allocation of buffer/bandwidth capacities would result in delivering a target QoS – provided that it is achievable – which is common to identical supported applications, such allocation will be ineffective when diverse QoS is to be delivered to applications with diverse traffic characteristics. Sophisticated buffer/bandwidth management schemes are needed to provide for the desired diversification, such as space priority for buffer management and priority scheduling for bandwidth allocation [3].

The work presented in this paper investigates the above mentioned problem of resource management at both time scales when the supported applications are diverse in terms of both the required QoS and traffic characteristics. In addition, the QoS requirement is assumed not to allow for packet (information unit) buffering beyond the present service cycle and, thus, buffer management is not considered to be an issue. Packets that do not receive service over the cycle following their arrival time are considered to have excess delay and are dropped. Applications with such QoS requirements are considered to be un-buffered and, thus, the problem is concentrated on the management of the available transmission resources at both time scales.

A wireless networking environment supporting real-time applications has characteristics of a shared un-buffered environment in which the management of the transmission resource is most critical due to its scarcity. Since the associated QoS requirements and the mobility of the supported users do not seem to allow for buffer management schemes, the work in this paper is presented in the context of wireless networking to provide for some reasonable motivation. The investigated problem of the achievable QoS and the associated scheduling schemes do maintain their general appeal throughout the paper.

Two major wireless networking technologies have recently emerged: cellular systems and wireless Local Area Networks (LAN). Cellular systems – originally designed to support voice applications – are currently facing the challenge of accommodating data and video services. Similarly, wireless LANs – developed to support computer data applications – are currently being redesigned to be capable of providing service to image, video and possibly voice applications. The convergence of the services supported by these two technologies drives a demand for a single network architecture capable of supporting all aforementioned services in an integrated fashion. In addition, this new architecture should be compatible with the prevalent network architecture for integrated services over fiber/copper based channels, the Asynchronous Transfer Mode (ATM). The development of an integrated services wireless network technology generates new transmission resource allocation problems that are more challenging than in the "specialized services" cellular voice systems or wireless LANs.

Consider for instance the transmission resource management problem in a cellular (voice) system. Since the supported (voice) applications are identical – in terms of traffic characteristics and QoS requirements – the transmission resource management problem at the large time scale is relatively easily addressed. Given a certain amount of available resources, the maximum number of voice applications that can be supported is such that the induced (*common*) system packet dropping *probability*, p_S, be less than a threshold, typically set at $p_S \leq 0.01$ [5]. Equivalently, this number may be determined as the one inducing a system packet dropping *rate*, $d_S \leq (0.01)(\lambda_S)$; $d_S(\lambda_S)$ is defined to be the average number of packets dropped (arrived) per service cycle. This simple rule also addresses the problem of determining the number of required transmission resources in order to support (simultaneously) a target number of applications. The uniformity in the QoS requirements and traffic characteristics of the supported voice applications simplifies substantially the transmission resource problem at the smaller time scale as well. A uniform, non-discriminating scheduling policy would induce the same (system) packet dropping probability to all supported applications.

The above discussion on resource management for cellular voice systems can be extended to that for wireless LANs. The QoS requirements of the supported computer applications suggest that resource management at the smaller time scale is of primary importance. Again, due to the uniformity of the QoS requirements, the resource management at the smaller time scale is not nearly as challenging as that in a integrated services network.

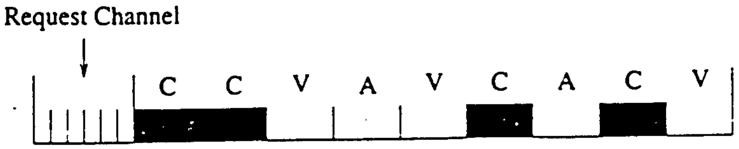
In this paper, the shared transmission resources are defined to be the slots (packet transmission times) of a TDMA frame. This resource structure has been widely considered in both cellular systems [7] and wireless LANs [1], as well as in recent work toward the development of wireless ATM networks [10,6]. *Variable Bit Rate* (VBR) applications with distinct QoS requirements and traffic characteristics are considered. The region of achievable multi-dimensional (non-degenerate) QoS is precisely determined and simple scheduling policies that deliver the achievable QoS are developed.

In the next section, the system model considered in this work is described. A set of equality and inequality constraints are presented in Section 3.1 that are employed in the establishment of the region of the achievable QoS vectors in Section 3.2. The derivations of Section 3 are employed in source call admission control and resource allocation applications and presented in Section 4. The special case of a homogeneous system is also presented in Section 4. Based on the developments of Section 3, a class of simple scheduling policies that achieve *any* achievable QoS vector is developed in Section 5. Some simple numerical examples are presented in Section 6, followed by conclusion of this work in Section 7.

2. Description of the system model

In this paper, the problem of sharing T resources by N diverse Variable Bit Rate (VBR) source applications is considered. The source packet arrival process is described in terms of a general arrival process embedded at the boundaries of fixed length intervals called service cycles (or frames). No additional assumptions for the packet arrival process are necessary at this point. Up to T packets may be transmitted (served) during each service cycle. Packets which cannot be transmitted over the service cycle following their arrival are considered to have excess delay and are dropped at the source.

The above environment could model the sharing of the up-link TDMA frame of a wireless network by N applications; T slots of this frame are available to these applications. Figure 1 illustrates a typical up-link TDMA frame whose slots are allocated to various classes of services, such as Continuous Bit Rate (CBR), Available Bit Rate (ABR) and Variable Bit Rate (VBR). In this figure $T = 3$ slots are assumed to be available to the VBR applications. Let $\lambda_i(n)$, $0 \leq i \leq N$, denote the number of source i packets generated at the nth frame boundary and requiring service over this frame. If $\sum_{i=1}^{N} \lambda_i(n) \leq T$ frame n is said to be *underloaded*; it is said to be *overloaded* if $\sum_{i=1}^{N} \lambda_i(n) > T$. Let $a_i^f(n)$, $0 \leq i \leq N$, denote the number of source i packets serviced during the nth frame under some scheduling policy f.



C	Assigned CBR slot
V	Slot available for VBR traffic
A	Slot available for ABR traffic

Fig. 1. TDMA frame supporting CBR, VBR and ABR traffic classes.

The number of source i packets dropped during frame n under policy f is given by $d_i^f(n)$, where,

$$d_i^f(n) = \lambda_i(n) - a_i^f(n) \begin{cases} = 0 & \text{if } \sum_{i=1}^{N} \lambda_i(n) \leq T \\ \geq 0 & \text{if } \sum_{i=1}^{N} \lambda_i(n) > T \end{cases}, \quad 0 \leq i \leq N. \tag{1}$$

Let $d_i^f = E[d_i^f(n)]$, $a_i^f = E[a_i^f(n)]$ and $\lambda_i = E[\lambda_i(n)]$ be the (assumed time invariant) expected values of the associated quantities.

Suppose that the QoS requirement of application i is defined in terms of a maximum tolerable per frame packet dropping rate d_i, $0 \leq i \leq N$. Then the QoS vector associated with the supported applications can be defined in terms of the (performance) packet dropping rate vector $\mathbf{d}$,

$$\mathbf{d} = (d_1, d_2, \ldots, d_N). \tag{2}$$

When the QoS requirement of application i is defined in terms of a maximum tolerable packet dropping probability p_i, the corresponding packet dropping rate d_i is easily determined by $d_i = \lambda_i \; p_i$.

The first question addressed in the sequel (Section 3) is whether a given QoS vectors $\mathbf{d}$ is achievable under any policy f. Necessary and sufficient conditions are derived in order for the QoS vector to be achievable, leading to the precise determination of the region of achievable QoS vectors $\mathbf{d}$. The second question, addressed in Section 4, is concerned with the design of scheduling policies which deliver an achievable QoS vector $\mathbf{d}$.

3. Determination of the region of achievable QoS vectors

In Section 3.1, a set of equality and inequality constraints associated with the packet dropping rate vector $\mathbf{d}^f$ induced by scheduling policies f are presented. These constraints lead to the derivation of the region of the achievable QoS vectors, presented in Section 3.2.

3.1. Conservation law and inequality constraints

Definition 3.1. A scheduling policy f, is work-conserving iff it satisfies the following conditions:

$$\sum_{i=1}^{N} d_i^f(n) = 0 \qquad \text{when} \sum_{i=1}^{N} \lambda_i(n) \le T \qquad \text{(underloaded frame)}, \tag{3}$$

$$\sum_{i=1}^{N} a_i^f(n) = T \qquad \text{when} \sum_{i=1}^{N} \lambda_i(n) > T \qquad \text{(overloaded frame)}. \tag{4}$$

A work-conserving policy does not waste resources (slots) as long as there is work to perform (packets to transmit). Let $S = \{1, 2, \dots, N\}$ be the set of all sources. Let $d_{\{S\}}^f$ denote the average system packet dropping rate under scheduling policy f, given by,

$$d_{\{S\}}^f \triangleq E\left[\sum_{i=1}^{N} d_i^f(n)\right] = \sum_{i=1}^{N} E\left[d_i^f(n)\right] = \sum_{i=1}^{N} d_i^f. \tag{5}$$

The following theorem provides for a well know result for work-conserving policies [4].

Theorem 3.1 (Conservation Law). *The system dropping rate, $d_{\{S\}}^f$, is conserved under any work-conserving policy f; that is,*

$$d_{\{S\}}^f \triangleq \left\{E\left[\sum_{i\in S} \lambda_i(n) \,\middle|\, \sum_{i\in S} \lambda_i(n) > T\right] - T\right\} P\left(\sum_{i\in S} \lambda_i(n) > T\right) \triangleq b_{\{S\}} \qquad \forall f. \tag{6}$$

Notice that the system dropping rate is independent of the policy f and only depends on the arrival process and the amount of resources T.

Proof. The number of source i packets dropped during frame n under policy f is given by,

$$d_i^f(n) = \lambda_i(n) - a_i^f(n). \tag{7}$$

Summing (7) over all sources i in the system S yields,

$$\sum_{i\in S} d_i^f(n) = \sum_{i\in S} \left(\lambda_i(n) - a_i^f(n)\right) = \sum_{i\in S} \lambda_i(n) - \sum_{i\in S} a_i^f(n). \tag{8}$$

Assuming f is work-conserving and using Eqs. (3) and (4) of Definition 3.1, the previous equation becomes,

$$\sum_{i\in S} d_i^f(n) = \begin{cases} 0 & \sum_{i\in S} \lambda_i(n) \le T \\ \sum_{i\in S} \lambda_i(n) - T & \sum_{i\in S} \lambda_i(n) > T. \end{cases} \tag{9}$$

By considering the expected value of (9) the proof of the theorem is completed. □

Let $d_{\{g\}}^f$ denote the average subsystem g packet dropping rate under policy f, defined by,

$$d_{\{g\}}^f = E\left[\sum_{i\in g} d_i^f(n)\right] = \sum_{i\in g} E\left[d_i^f(n)\right] = \sum_{i\in g} d_i^f, \quad g \subseteq S = \{1, 2, \dots, N\}. \tag{10}$$

That is, $d^f_{\{g\}}$ is equal to the cumulative packet dropping rate associated with sources in group g only, under policy f; all N sources in S are assumed to be present and served under policy f.

Let $b_{\{g\}}$ denote the lower bound for the cumulative packet dropping rate for sources in g, determined to be equal to the packet dropping rate of a system in which only sources in g are present and served under a work-conserving policy. Sources in set $\{S-g\}$ are considered to be removed. In view of Theorem 3.1 and by applying Eq. (6) for the subsystem g, the following is derived,

$$b_{\{g\}} = \left\{ E\left[\sum_{i\in g} \lambda_i(n) \,\middle|\, \sum_{i\in g} \lambda_i(n) > T \right] - T \right\} \Pr\left(\sum_{i\in g} \lambda_i(n) > T \right), \quad g \subseteq S. \tag{11}$$

The following theorem provides for a set of inequality constraints associated with scheduling policies.

Theorem 3.2. *The following constraints associated with the induced packet dropping rate vector* $\mathbf{d}^f = (d^f_1, d^f_2, \ldots, d^f_N)$ *are satisfied by any scheduling policy:*

$$d^f_{\{g\}} \geq b_{\{g\}}, \quad \forall g \subset S = \{1, 2, \ldots, N\}, \tag{12}$$

where $d^f_{\{g\}}$ *and* $b_{\{g\}}$ *are given in (10) and (11).*

Proof. Suppose that the packets from sources in g are served under a work-conserving scheduling policy f_o in which they are given Head of the Line (HoL) priority over packets from sources in $\{S-g\}$. That is, no packet from sources in $\{S-g\}$ is served unless no packets from sources in g are present. As a consequence, packets from sources in g do not experience interference from the traffic delivered by sources in $\{S-g\}$ and thus,

$$\sum_{i\in g} d^{f_o}_i(n) = 0 \quad \text{when} \sum_{i\in g} \lambda_i(n) \leq T, \tag{13}$$

$$\sum_{i\in g} a^{f_o}_i(n) = T \quad \text{when} \sum_{i\in g} \lambda_i(n) > T. \tag{14}$$

From (10) and (11) it is clear that,

$$d^{f_o}_{\{g\}} = b_{\{g\}}, \tag{15}$$

for all work-conserving policies that provide HoL priority to the packets from sources in g. Since no other policy f can provide better service (that is lower packet dropping rates) to sources in g than f_o, it is evident that,

$$d^f_{\{g\}} \geq b_{\{g\}}, \quad \forall f,\ g \subset S. \quad \square \tag{16}$$

3.2. Achievable region for QoS vectors

The main result of this section is the determination of the region $\mathcal{D}$ of the achievable QoS vectors $\mathbf{d}$. The following corollary provides a set of necessary conditions in order for a QoS vector $\mathbf{d}$ to be achievable, followed by a corollary regarding an upper bound on the achievable region for QoS vectors $\mathbf{d}$. Their proofs are self-evident in view of Theorems 3.1 and 3.2.

Corollary 3.1. *A necessary condition in order for a QoS vector* $\mathbf{d} = (d_1, d_2, \ldots, d_N)$ *to be achieved by a work-conserving scheduling policy* f *is that its components satisfy the following constraints,*

$$d_{\{g\}} \geq b_{\{g\}} \quad \forall g \subset S, \tag{17}$$

$$d_{\{S\}} = b_{\{S\}}. \tag{18}$$

Corollary 3.2. *Let* $\mathcal{D}^u$ *denote the collection of all vectors* $\mathbf{d}$ *satisfying (17) and (18). Then* $\mathcal{D}^u$ *is an upper bound on the region* $\mathcal{D}$ *of achievable QoS vectors* $\mathbf{d}$. *That is,*

$$\mathcal{D} \subseteq \mathcal{D}^u. \tag{19}$$

The following theorem describes the vectors contained in $\mathcal{D}^u$.

Theorem 3.3. *Any vector in the set* $\mathcal{D}^u$ *can be expressed as a convex combination of extreme points (vertices) of* $\mathcal{D}^u$*; that is,* $\mathcal{D}^u$ *may be expressed as the convex hull of its extreme points,* $\mathcal{D}^u = \mathrm{conv}[ext(\mathcal{D}^u)]$.

Proof. The proof follows from the fact that $\mathcal{D}^u$ is a bounded set defined by a finite intersection of closed half spaces, (see (17) and (18)). Then by definition, $\mathcal{D}^u$ is a polytope [9] and Theorem 3.3 follows directly from properties of polytopes [9]. □

The following theorem establishes a relationship between scheduling policies and the vertices of $\mathcal{D}^u$.

Theorem 3.4. $\mathbf{d}^*$ *is a vertex of the set* $\mathcal{D}^u$ *iff* $\mathbf{d}^*$ *is a dropping rate vector resulting from an Ordered HoL (O-HoL) priority service policy,* $\pi = (\pi_1, \pi_2, \ldots, \pi_N)$; $\pi_i \in \{1, 2, \ldots, N\}$, $\pi_i \neq \pi_j$, $1 \leq i, j \leq N$. *The index of* π_i *indicates the order of the priority given to the* π_i *source. None of the* π_j *sources,* $j > i$, *may be served as long as packets from sources* π_k, $k \leq i$, *are present.*

Proof. Assume that $\mathbf{d}^*$ is a vertex of $\mathcal{D}^u$. Then $\mathbf{d}^*$ must lie at the intersection of N hyper-planes and its coordinates must satisfy N simultaneous, linear independent equations (definition of a polytope vertex, see [9]), given by,

$$\sum_{i \in g_j} d_i^* = b_{\{g_j\}}, \quad j = 1, 2, \ldots, N, \tag{20}$$

where one of the g_j's is the set $S = \{1, 2, \ldots, N\}$ and the remaining $(N-1)$ are proper, non-empty and different subsets of S. Lemma A.2 (see appendix) establishes that the subsets g_i's are strictly included in each other. Therefore, by adopting the order $g_1 \subset g_2 \subset \cdots \subset g_{N-1} \subset g_N \equiv S$, the g_i's are given by,

$$\begin{aligned} g_1 &= \{\pi_1\} \\ g_1 \subset g_2 &= \{\pi_1, \pi_2\} \\ g_2 \subset g_3 &= \{\pi_1, \pi_2, \pi_3\} \\ &\vdots \\ g_{N-1} \subset g_N &= \{\pi_1, \pi_2, \ldots, \pi_N\} = \{1, 2, \ldots, N\}, \end{aligned} \tag{21}$$

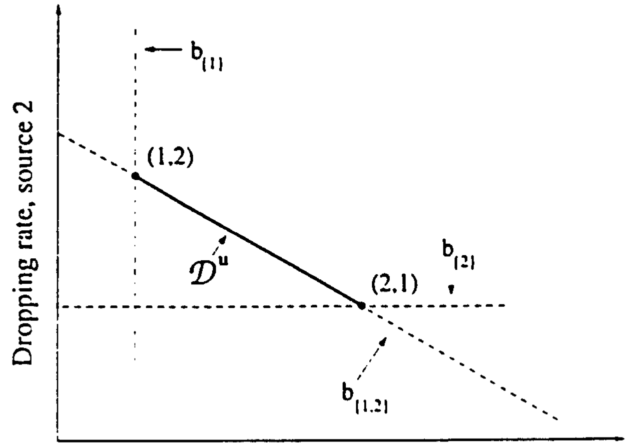


Fig. 2. The region (polytope) $\mathcal{D}^u$ for a system with two sources.

where $\pi_i \in \{1, 2, \dots, N\}$, $\pi_i \neq \pi_j$, for $i \neq j$, $1 \leq i$, $j \leq N$. Substituting into (20) yields,

$$\begin{aligned} d^*_{\{\pi_1\}} &= b_{\{\pi_1\}} \\ d^*_{\{\pi_2\}} &= b_{\{\pi_1,\pi_2\}} - b_{\{\pi_1\}} \\ d^*_{\{\pi_3\}} &= b_{\{\pi_1,\pi_2,\pi_3\}} - b_{\{\pi_1,\pi_2\}} \\ &\vdots \\ d^*_{\{\pi_N\}} &= b_{\{\pi_1,\pi_2,\dots,\pi_N\}} - b_{\{\pi_1,\pi_2,\dots,\pi_{N-1}\}}, \end{aligned} \tag{22}$$

which is precisely the dropping rate vector induced by the ordered HoL priority policy $\pi = (\pi_1, \pi_2, \dots, \pi_N)$. Thus for any vertex there exists an ordered HoL priority policy that induces it. Since it is easy to see that the dropping rate vector resulting from an ordered HoL priority policy must satisfy (22), it is evident that these dropping rate vectors will be vertices of $\mathcal{D}^u$. □

Figures 2 and 3 provide a graphical illustration of the region $\mathcal{D}^u$ for the case of $N = 2$ and $N = 3$ sources, respectively. The extreme points $\mathbf{d}_{ext-i}$'s correspond to QoS vectors induced by the $N!$ ordered HoL priority policies $\pi^i = (\pi_1, \pi_2, \dots, \pi_N)$, $1 \leq i \leq N!$, as shown in Theorem 3.4. Referring to Fig. 2, it may be observed that the policy $(\pi_1, \pi_2) = (1, 2)$ corresponds to the intersection of the line for the lower bound on the packet dropping rate line for source 1, $b_{\{1\}}$, with the system dropping rate line $b_{\{1,2\}}$. Similarly, the second extreme point induced by the policy $(\pi_1, \pi_2) = (2, 1)$ is the intersection of the lower bound on the packet dropping rate for source 2, $b_{\{2\}}$ and $b_{\{1,2\}}$. Similar observations can be made for the region $\mathcal{D}^u$ for a system of $N = 3$ sources shown in Fig. 3.

Let $\mathcal{D}$, $\mathcal{D} \subseteq \mathcal{D}^u$, be the achievable region of QoS vectors $\mathbf{d}$. The following theorem establishes its convexity.

Theorem 3.5. *Let* $\mathbf{d}_1, \mathbf{d}_2 \in \mathcal{D}$, *then* $\mathbf{d}_3$, *where*

$$\mathbf{d}_3 = \alpha \mathbf{d}_1 + (1 - \alpha)\mathbf{d}_2, \qquad \alpha \geq 0,\ \alpha \leq 1, \tag{23}$$

is also in $\mathcal{D}$. *That is* $\mathcal{D}$ *is convex.*

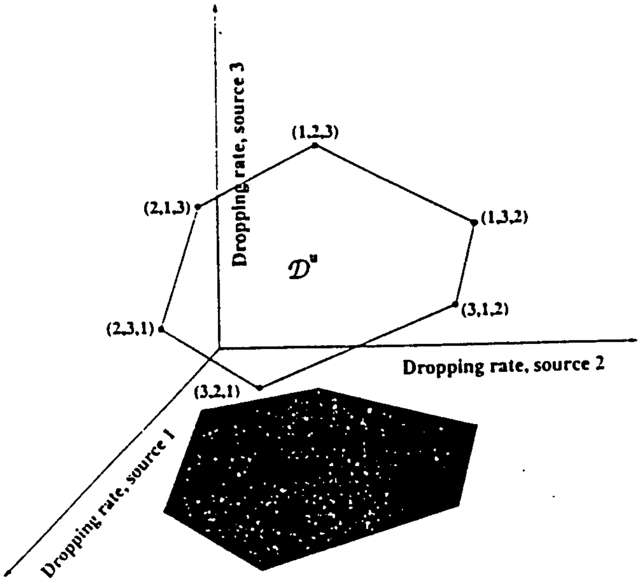


Fig. 3. The region (polytope) $\mathcal{D}^u$ for a system with three sources.

Proof. Let $\mathbf{d}_1$ ($\mathbf{d}_2$) be the dropping rate vector induced the achievable policy S_1 (S_2); that is, $\mathbf{d}_1, \mathbf{d}_2 \in \mathcal{D}$. Consider a scheduling policy S_3 that at each frame decides to follow the scheduling rule of policy S_1 with probability α and policy S_2 with probability $(1-\alpha)$; the decisions over consecutive frames are independent. The packet dropping rate induced by policy S_3 is given by,

$$\mathbf{d}_3 = \alpha\mathbf{d}_1 + (1-\alpha)\mathbf{d}_2. \tag{24}$$

Since $\mathbf{d}_3$ is achieved by a policy, $\mathbf{d}_3 \in \mathcal{D}$, establishing the convexity of $\mathcal{D}$. □

The next theorem establishes the region $\mathcal{D}$ of achievable QoS vectors.

Theorem 3.6. $\mathcal{D} \equiv \mathcal{D}^u$.

Proof. Since $\mathcal{D} \subseteq \mathcal{D}^u$ (Corollary 3.2), it suffices to establish that $\mathcal{D}^u \subseteq \mathcal{D}$ to complete the proof. Notice that $\mathcal{D}^u = \text{conv}[ext(\mathcal{D}^u)]$ (Theorem 3.3) and that $\mathcal{D}$ is convex (Theorem 3.5) and, thus, if $\{ext(\mathcal{D}^u)\} \subseteq \mathcal{D}$ then $\mathcal{D}^u \subseteq \mathcal{D}$. The latter holds true since the extreme points of $\mathcal{D}^u$ are induced by the ordered HoL priority policies (Theorem 3.4) and, thus, these points are in the region $\mathcal{D}$ of achievable QoS vectors. □

4. Applications

In the proceeding sections a system of N sources sharing T resources has been considered and the region $\mathcal{D}$ of achievable QoS vectors $\mathbf{d}$ induced by any work-conserving policy has been established. If the amount of available resources T is fixed, it is important to determine whether all N diverse applications can be supported. If the required resources can be made available, it is important to determine the minimum number of required resources. These two questions are addressed in this section.

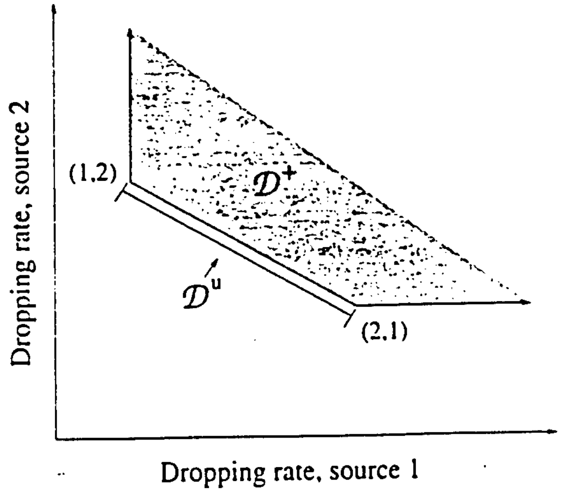


Fig. 4. Achievable region $\mathcal{D}$ and call-admission region (polyhedral) $\mathcal{D}^+$ for a system with two sources.

4.1. Call admission control

The following definitions facilitates the discussion of this section.

Definition 4.1. The *worst-tolerable* QoS-vector $\hat{\mathbf{d}} = (\hat{d}_1, \hat{d}_2, \ldots, \hat{d}_N)$ associated with the N applications is defined to be the one whose coordinates are equal to the largest packet dropping rate acceptable by the corresponding application.

Definition 4.2. $\mathbf{d}^1 = (d_1^1, d_2^1, \ldots, d_N^1)$ is said to be *better* than $\mathbf{d}^2 = (d_1^2, d_2^2, \ldots, d_N^2)$ — denoted by $\mathbf{d}^1 \leq \mathbf{d}^2$ — iff

$$d_i^1 \leq d_i^2, \quad 1 \leq i \leq N. \tag{25}$$

Definition 4.3. The QoS of N applications is said to be *satisfied* if there exists a policy inducing a packet dropping rate vector $\mathbf{d}$ (that is $\mathbf{d} \in \mathcal{D}$) which is better than the worst-tolerable QoS vector associated with these applications.

Definition 4.4. Let $b_{\{g\}}$, $g \subseteq S = \{1, 2, \ldots, N\}$ be the lower bound on the cumulative packet dropping rates associated with subsets of N given application and T available resources ($b_{\{g\}}$ is defined precisely earlier); let $\mathcal{D}$ be the associated region of achievable QoS vectors. The *call admission region* $\mathcal{D}^+$ is defined to be the set of dropping rate vectors $\mathbf{d} = (d_1, d_2, \ldots, d_N)$ satisfying,

$$d_{\{g\}} \geq b_{\{g\}}, \quad \forall g \subseteq S. \tag{26}$$

Notice that for $g \equiv S$, (26) is satisfied with strict equality for the dropping rate vectors in $\mathcal{D}$. Thus, $\mathcal{D}^+$ is a polyhedral having $\mathcal{D}$ as one of its faces [9]. A graphical representation of the call acceptance region $\mathcal{D}^+$ is shown in Figs. 4 and 5 for a system of $N = 2$ and $N = 3$ sources, respectively.

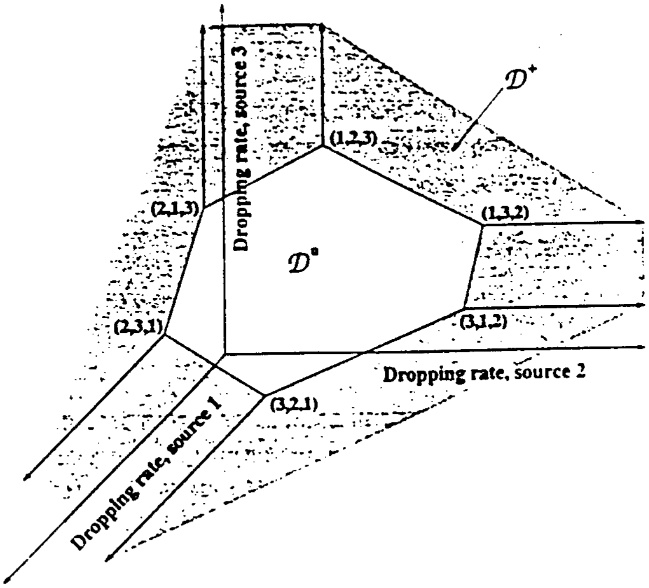


Fig. 5. Achievable region $\mathcal{D}$ and call-admission region (polyhedral) $\mathcal{D}^+$ for a system with three sources.

The following theorem establishes a call admission rule.

Theorem 4.1. *Let* $\hat{\mathbf{d}} = (\hat{d}_1, \hat{d}_2, \dots, \hat{d}_N)$ *be the worst-tolerable QoS vector associated with N given applications. Then if* $\hat{\mathbf{d}}$ *is in the call-acceptance region* $\mathcal{D}^+$*, then the QoS of the N applications can be satisfied.*

Proof. The proof is self-evident provided that there exists a policy that induces a dropping rate vector $\mathbf{d}'$, such that $\mathbf{d}' \in \mathcal{D}$ and $\mathbf{d}' \leq \hat{\mathbf{d}}$. The latter holds true if,

$$\begin{aligned} \hat{d}_{\{g\}} &\geq d'_{\{g\}} \geq b_{\{g\}} \qquad \forall g \subset S, \\ \hat{d}_{\{S\}} &\geq d'_{\{S\}} = b_{\{S\}}, \end{aligned} \tag{27}$$

which is established in [12] by denoting $\mathcal{D}$ to be the core of the game (S, b). According to [12] the core of game (S, b) consists of all vectors $\mathbf{d}$ which satisfy $d_{\{S\}} = b_{\{S\}}$ and $d_{\{g\}} \geq b_{\{g\}}$ for all $g \subset S$ where $d_{\{g\}} = \sum_{i \in g} d_i$; the core of the game is defined to be large if the conditions in (27) are satisfied for every $\hat{\mathbf{d}} \in \mathcal{D}^+$. A sufficient condition for the core of the game to be large is for the set function $b_{\{g\}}$ to be *super modular*, see Theorem 4 of [12]; this result has been established in the appendix, see Lemma A.1. □

4.2. Resource allocation

As implied earlier in the discussion associated with Fig. 1, applications in integrated services networks are not expected to be allowed to access *all* available resources. Some broad groups (or classes) are expected to be defined and only a proportion of the resources will typically be available to each group. If it is desirable that a certain number of N applications be supported, it is important that the minimum amount of required resources to be alloted to that group be determined. This problem is discussed below.

To emphasize the dependence of the quantities $\mathcal{D}, \mathcal{D}^+$ and $b_{\{g\}}$ for all $g \subseteq S$, on T, they will be written explicitly as functions of T and be written as $\mathcal{D}(T)$, $\mathcal{D}^+(T)$ and $b_{\{g\}}(T)$, respectively.

Theorem 4.2. *Let $\hat{\mathbf{d}} \doteq (\hat{d}_1, \hat{d}_2, \dots, \hat{d}_N)$ be the worst-tolerable QoS vector associated with N applications. The minimum amount of resources – denoted by T_{min} – required in order for the QoS of these applications to be satisfied is given by,*

$$T_{min} = \Big\{T,\ T \ge 1 : \Big[\Big(\hat{d}_{\{g\}} \ge b_{\{g\}}(T), \forall g \subseteq S\Big) \quad and \quad \Big(\hat{d}_{\{g_o\}} < b_{\{g_o\}}(T-1) \quad for\ some\ g_o \subseteq S\Big)\Big]\Big\}. \tag{28}$$

Proof. Since $\hat{\mathbf{d}} \in \mathcal{D}^+(T)$, where $\mathcal{D}^+(T)$ is the call acceptance region determined by T and $\{b_{\{g\}},\ g \subseteq S\}$, the QoS of the N applications will be satisfied (Theorem 4.1). To establish that T_{min} determined by (28) is indeed the minimum it suffices to show that $b_{\{g\}}(T)$ are non-decreasing functions of T. This is obvious since the lower bound on the dropping rates for a group g, $b_{\{g\}}$, cannot increase as the number of service opportunities fixed service cycle increase. □

4.3. The special case of homogeneous applications

In this section the theory developed earlier is applied to a system supporting identical applications. That is, both the packet arrival process and the largest acceptable packet dropping rates are identical. In this case, the worst-tolerable QoS vector will have identical coordinates; that is $\hat{\mathbf{d}} = (\hat{d}_v, \hat{d}_v, \dots, \hat{d}_v)$. The following theorems establish a (simplified) call admission rule for a homogeneous system.

Theorem 4.3. *Let $\hat{\mathbf{d}} = (\hat{d}_v, \hat{d}_v, \dots, \hat{d}_v)$ be the worst-tolerable QoS vectors associated with the N homogeneous applications. The vector $\hat{\mathbf{d}}$ is in the call-acceptance region $\mathcal{D}^+$ (and, thus, the QoS of all N applications is satisfied) if*

$$\hat{d}_S \ge b_S \quad or \quad \hat{d}_v \ge \frac{b_{\{S\}}}{N}. \tag{29}$$

Proof. The definition of $\mathcal{D}^+$ (Definition 4.1) and the call-admission rule (Theorem 4.1) establish that (29) is a necessary and sufficient condition provided that it implies that,

$$d_{\{g\}} \ge b_{\{g\}} \quad \text{or} \quad \hat{d}_v \ge \frac{b_{\{g\}}}{|g|}, \quad \forall g \subseteq S, \tag{30}$$

where $|g|$ denotes the number of elements (sources) in set g. In view of (29), (30) will hold if,

$$\frac{b_{\{S\}}}{N} \ge \frac{b_{\{g\}}}{|g|}, \quad \forall g \subseteq S. \tag{31}$$

Since all the sources are statistically identical, set functions $b_{\{g\}}$ associated with sets g having the same number of sources are equal. That is,

$$b_{\{a_1,\dots,a_m\}} = b_{\{1,\dots,m\}}, \quad \forall \{a_1, \dots, a_m\},\ m = 1, 2, \dots, N,\ a_i \ne a_j,\ a_i, a_j \in S. \tag{32}$$

Using (32) and Lemma A.1,

$$\begin{aligned} 2b_{\{a_1\}} &= b_{\{a_1\}} + b_{\{a_2\}} < b_{\{a_1,a_2\}} \\ 3b_{\{a_1,a_2\}} &= b_{\{a_1,a_2\}} + b_{\{a_1,a_3\}} + b_{\{a_2,a_3\}} < b_{\{a_1,a_2,a_3\}} + b_{\{a_1\}} + b_{\{a_2,a_3\}} < 2b_{\{a_1,a_2,a_3\}} \\ &\vdots \\ (|g|+1)b_{\{g\}} &< |g|b_{\{g\cup a\}} \qquad a \notin \{g\}. \end{aligned} \tag{33}$$

Equivalently,

$$\frac{b_{\{g\cup a\}}}{|g|+1} > \frac{b_{\{g\}}}{|g|} \qquad a \notin \{g\}, \tag{34}$$

and (31) immediately follows from this result.

5. A class of policies inducing all QoS vector in $\mathcal{D}$

Let C_{O-HoL} denote the class of ordered HoL priority service policies π^{i}'s introduced in Section 3.2.

Definition 5.1. A mixing ordered HoL priority service policy f_m is defined to be one that at each frame decides to follow the ordered HoL priority policy $\pi^{i} = (\pi_1^i, \pi_2^i, \ldots, \pi_N^i)$ with probability $\alpha_i, \alpha_i \geq 0$, $1 \leq i \leq N!$, $\sum_{i=1}^{N!} \alpha_i = 1$; decisions over consecutive frames are independent. Clearly, f_m is completely determined by the $N!$ dimensional vector $\boldsymbol{\alpha}$, $\boldsymbol{\alpha} \geq \mathbf{0}$, $\mathbf{1} \cdot \boldsymbol{\alpha} = 1$. Let M_{O-HoL} denote the class of such policies.

The following theorem establishes the main results of this section.

Theorem 5.1. *For each packet dropping rate vector* $\mathbf{d} \in \mathcal{D}$ *there exist a policy* $f_m \in M_{O-HoL}$ *that induces* $\mathbf{d}$.

Proof. Let $\mathbf{d} \in \mathcal{D}$. Then $\mathbf{d} = \sum_{i=1}^{N!} \alpha_i \mathbf{d}_{ext-i}$ for some $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_{N!})$ where $\alpha_i \geq 0$, $1 \leq i \leq N!$, $\sum_{i=1}^{N!} \alpha_i = 1$, since any point in $\mathcal{D}$ can be written as a convex combination of the extreme points (vertices) $\mathbf{d}_{ext-i}$ of $\mathcal{D} (= \mathcal{D}^u)$; each $\mathbf{d}_{ext-i}$ is induced by some policy in C_{O-HoL} (Theorem 3.4).

Let f_m be the mixing policy which selects the HoL priority π^{i} (that induces $\mathbf{d}_{ext-i}$) with probability α_i. The packet dropping rate vector $\mathbf{d}^{f_m}$ induced by f_m is given by,

$$\mathbf{d}^{f_m} = \sum_{i=1}^{N!} \alpha_i \mathbf{d}_{ext-i} \tag{35}$$

and thus f_m induces $\mathbf{d}$. □

The following corollary is obvious in view of the above theorem.

Corollary 5.1. *Let* $\mathbf{d} \in \mathcal{D}$ *be a target packet dropping rate vector. The mixing policy* $f_m \equiv \boldsymbol{\alpha}$ *induces* $\mathbf{d}$, *where* α *is such that,*

$$E_{\alpha}[\mathbf{d}_{ext}] \triangleq \sum_{i=1}^{N!} \alpha_i \mathbf{d}_{ext-i} = \mathbf{d}, \tag{36}$$

$$\alpha \geq 0, \tag{37}$$

$$\mathbf{1} \cdot \alpha = 1, \tag{38}$$

where $E_{\alpha}[.]$ *is weighted average of the set of extreme points,* $\mathbf{d}_{ext}$*, of* $\mathcal{D}$ *with respect to the Probability Mass Function* α.

6. A numerical example

In this section a numerical example is carried out for a system with three sources competing for T slots in a TDMA frame. The source packet arrival processes are assumed to be mutually independent. Each arrival process is described in terms of an underlying finite-state Markov chain embedded at the frame boundaries. The number of packets generated (and requesting service) by a source in the current frame boundary is (probabilistically) determined by the present state of the associated underlying Markov chain.

Consider in this example the typical TDMA frame in which each slot represents a 32 Kbps channel. Let $E^k = \{0, 1, \ldots, M^k - 1\}$ denote the state space of the Markov chain associated with source k, $k = 1, 2, 3$. In this example $M^1 = 2$, $M^2 = 3$ and $M^3 = 4$. When in state i, a source generates $2i$ packets in the current frame. Source 1, has a maximum transmission rate of 64 Kbps, it can model an uncompressed voice source [2]. Source 2 is the superposition of two uncompressed voice sources. Source 3 could be used to model a low rate video transmission with maximum rate 6 slots/frame = 192 Kbps and step size of 64 Kbps. Further description of all three sources, including transition rates, can found in Tables 1, 2 and 3; α_i^k (β_i^k) denote the transition rates from state i to state $i + 1$ (state i to $i - 1$), of source k; i, $i - 1$, $i + 1 \in E^k$, $k = 1, 2, 3$. The per frame arrival rates are easily computed and are equal to $\lambda_1 = 0.8$, $\lambda_2 = 1.6$ and $\lambda_3 = 4.0$ packets per frame.

Let p_k denote the target packet dropping probability, for source k, $k = 1, 2, 3$. In this example it is assumed that $p_1 = 0.02$, $p_2 = 0.04$ and $p_3 = 0.02$. By multiplying these probabilities by the corresponding per frame arrival rate, the per frame packet dropping rates $d_1 = 0.016$, $d_2 = 0.064$ and $d_3 = 0.08$ are obtained. Thus, the worst-tolerable QoS vector $\mathbf{d}$ is given by $\mathbf{d} = (d_1, d_2, d_3) = (0.016, 0.064, 0.08)$ and the associated system packet dropping rate is equal to $d_{\{S\}} = d_1 + d_2 + d_3 = 0.160$. By applying Theorem 4.2 it is established that the minimum amount of resources required in order for $\mathbf{d}$ to be in the call-acceptance region are $T_{min} = 9$ slots per TDMA frame.

Table 1

Description of source 1

State, i	α_i^1	β_i^1	State probability
0	0.667	–	0.6
1	–	1.0	0.4

Table 2

Description of source 2

State, i	α_i^2	β_i^2	State probability
0	1.330	–	0.36
1	0.667	1	0.48
2	–	2	0.16

Table 3
Description of source 3

State, i	α_i^3	β_i^3	State probability
0	2.7000	–	0.05
1	4.4000	0.9	0.15
2	0.6818	1.2	0.55
3	–	1.5	0.25

In this case $d_{\{S\}} \geq b_{\{S\}} = 0.1552$ and any vector $\mathbf{d}'$ satisfying conditions in Theorem 4.1 can be selected. The (non-unique) vector $\mathbf{d}' = (0.016, 0.064, 0.0752)$ was found by requiring (last constraint in Theorem 4.1) that $d'_{\{S\}} = b_{\{S\}} = 0.1552$. Since $\mathbf{d}' \in \mathcal{D}$, there exists a mixing ordered HoL priority policy $f_m \equiv \boldsymbol{\alpha}$ achieving exactly the QoS vector $\mathbf{d}'$. Any $\boldsymbol{\alpha}$ satisfying the conditions of Corollary 5.1, may be chosen. For this example and employing linear programming techniques, the following $\boldsymbol{\alpha}_o$ was found,

$$\boldsymbol{\alpha}_o = \begin{bmatrix} \alpha_1 = 0.1280 \\ \alpha_2 = 0.0000 \\ \alpha_3 = 0.2843 \\ \alpha_4 = 0.1810 \\ \alpha_5 = 0.2678 \\ \alpha_6 = 0.1389 \end{bmatrix} \tag{39}$$

As it is expected from the formulation of the constraints, more than one solution may be found. This allows for the incorporation of additional constraints representing other desirable qualities of the policies. Functions of interest may be minimized subject to the constraints presented in this paper to guarantee the achievability of the resulting policies. For instance, among all mixing policies inducing $\mathbf{d}'$, the one which minimizes the variance of the service provided to certain sources may be identified. Such additional objectives will be pursued in the future.

To further illustrate the design concept, consider the following two user system. In this example $M^1 = 3$, $M^2 = 4$. As in the first example, when in state i a source generates $2i$ packets in the current frame. Source 1 and source 2 have probability state distributions, (0.1, 0.3, 0.6) and (0.5, 0.3, 0.1, 0.1), respectively. The per frame arrival rates are easily computed and are equal to $\lambda_1 = 3.0$ and $\lambda_2 = 1.6$ packets per frame.

In this example the worst-tolerable QoS vector $\mathbf{d}$ is given by $\mathbf{d} = (d_1, d_2) = (0.781, 0.099)$ and the associated system packet dropping rate is equal to $d_{\{S\}} = d_1 + d_2 = 0.790$. In this example, the system packet dropping rate can be satisfied with $T = 5$ slots, that is $d_{\{S\}} = b_{\{S\}} = 0.790$. Although with $T = 5$, $d_2 < b_2 = 0.10$, and thus the worst-tolerable QoS vector is not achievable. In this case the best possible performance that source 2 can receive is if it is given priority, resulting in dropping rates of 0.780 and 0.10 for source 1 and 2 respectively. The overall system performance is satisfied, but source 2 is experiencing poorer service than its worst-tolerable, while source 1 is experiencing improved performance. Applying Theorem 4.2 it is established that the minimum amount of resources required in order for $\mathbf{d}$ to be in the call-acceptance region are $T_{min} = 6$ slots per TDMA frame.

7. Conclusion

In this work the region of achievable QoS vectors for an integrated services wireless network was precisely determined. The QoS in a shared un-buffered environment (wireless network) is measured by the packet dropping probability, p_i. The determination of the region of achievability for this environment was first made possible by noticing that a conservation law applies to the packet dropping rate, d_i, from which a packet dropping probability can be inferred as $p_i = \frac{d_i}{\lambda_i}$. Secondly, the super-modularity of the set function $b_{\{g\}}$, describing the lower bound on the cumulative packet dropping rate, made it possible to classify the extreme points (vertices) of the region of achievability.

Knowledge of the region of achievable QoS vectors led to the development of an admission control rule and resource allocation techniques for the applications in the integrated services wireless network. Admission control for a special case of a homogeneous system was considered. The anticipated result, $d_v \geq \frac{b_{\{S\}}}{N}$, or equivalently, $p_i \geq \frac{b_{\{S\}}}{N\,\lambda_v}$, which had been seen in [7] as a sufficient condition for admission to a cellular voice network, was derived.

Once it was established that the target QoS vector was in the achievable region, easy to implement scheduling policies were derived that achieve the desired result. It was shown that there exists a class of scheduling policies – mixed Ordered HoL policies – that can achieve any achievable QoS vector.

Appendix A

Lemma A.1. *b_A is a super-modular set function, that is,*

$$b_A + b_B \leq b_{\{A\cup B\}} + b_{\{A\cap B\}} \tag{40}$$

and equality hold only if $A \subset B$ or $B \subset A$.

Proof. Let λ_A (λ_B) be the generic random variables representing the number of packets generated by sources in set A (B) over a frame, $A, B, \subseteq S$. Clearly,

$$\lambda_A + \lambda_B = \lambda_{\{A\cup B\}} + \lambda_{\{A\cap B\}} \qquad \forall A, B \subseteq S. \tag{41}$$

Case (I): Assume A and B are disjoint ($A \cap B = \emptyset$). In this case $\lambda_{\{A\cup B\}} = \lambda_A + \lambda_B$ and,

$$b_{\{A\cup B\}} = \left\{ E\left[\lambda_A + \lambda_B \,\middle|\, \lambda_A + \lambda_B > T\right] - T \right\} \Pr(\lambda_A + \lambda_B > T). \tag{42}$$

The following implication is self-evident:

$$\lambda_A + \lambda_B > T \Longrightarrow \begin{cases} \lambda_A > T, \lambda_B \leq T \\ \lambda_A \leq T, \lambda_B > T \\ \lambda_A > T, \lambda_B > T \\ \lambda_A \leq T, \lambda_B \leq T, \lambda_A + \lambda_B > T. \end{cases} \tag{43}$$

Assuming that all sources are independent then λ_A and λ_B are independent since sets A and B are disjoint, then,

$$\begin{aligned}\Pr(\lambda_A + \lambda_B > T) = {} & \Pr(\lambda_A > T)\, P(\lambda_B \leq T) + \Pr(\lambda_A \leq T)\, P(\lambda_B > T) \\ & + \Pr(\lambda_A \leq T, \lambda_B \leq T, \lambda_A + \lambda_B > T) + \Pr(\lambda_A > T)\, P(\lambda_B > T),\end{aligned} \tag{44}$$

therefore (42) may be written as,

$$\begin{aligned} b_{\{A\cup B\}} = & \left\{E\left[\lambda_A+\lambda_B \,\middle|\, \lambda_A > T, \lambda_B \le T\right] - T\right\} \Pr(\lambda_A > T) \Pr(\lambda_B \le T) \\ & + \left\{E\left[\lambda_A+\lambda_B \,\middle|\, \lambda_A \le T, \lambda_B > T\right] - T\right\} \Pr(\lambda_A \le T) \Pr(\lambda_B > T) \\ & + \left\{E\left[\lambda_A+\lambda_B \,\middle|\, \lambda_A > T, \lambda_B > T\right] - T\right\} \Pr(\lambda_A > T) \Pr(\lambda_B > T) \\ & + \left\{E\left[\lambda_A+\lambda_B \,\middle|\, \lambda_A \le T, \lambda_B \le T, \lambda_A+\lambda_B > T\right] - T\right\} \\ & \cdot \Pr(\lambda_A \le T, \lambda_B \le T, \lambda_A+\lambda_B > T). \end{aligned} \tag{45}$$

Expanding the expected value operator over λ_A and λ_B and rearranging terms, (45) can be written as,

$$\begin{aligned} b_{\{A\cup B\}} = & \left\{E\left[\lambda_A \,\middle|\, \lambda_A > T\right] - T\right\} \Pr(\lambda_A > T) + \left\{E\left[\lambda_B \,\middle|\, \lambda_B > T\right] - T\right\} \Pr(\lambda_B > T) \\ & + E\left[\lambda_A \,\middle|\, \lambda_A \le T\right] \Pr(\lambda_A \le T) \Pr(\lambda_B > T) \\ & + E\left[\lambda_B \,\middle|\, \lambda_B \le T\right] \Pr(\lambda_A > T)\, P(\lambda_B \le T) + T \Pr(\lambda_A > T)\, P(\lambda_B > T) \\ & + \left\{E\left[\lambda_A+\lambda_B \,\middle|\, \lambda_A \le T, \lambda_B \le T, \lambda_A+\lambda_B > T\right] - T\right\} \\ & \cdot \Pr(\lambda_A \le T, \lambda_B \le T, \lambda_A+\lambda_B > T) \\ = & \; b_{\{A\}} + b_{\{B\}} + C^{\lambda_A,\lambda_B} \end{aligned} \tag{46}$$

where,

$$\begin{aligned} C^{\lambda_A,\lambda_B} = & \; E\left[\lambda_A \,\middle|\, \lambda_A \le T\right] \Pr(\lambda_A \le T) \Pr(\lambda_B > T) \\ & + E\left[\lambda_B \,\middle|\, \lambda_B \le T\right] \Pr(\lambda_A > T)\, P(\lambda_B \le T) + T \Pr(\lambda_A > T)\, P(\lambda_B > T) \\ & + \left\{E\left[\lambda_A+\lambda_B \,\middle|\, \lambda_A \le T, \lambda_B \le T, \lambda_A+\lambda_B > T\right] - T\right\} \\ & \cdot \Pr(\lambda_A \le T, \lambda_B \le T, \lambda_A+\lambda_B > T). \end{aligned}$$

Notice that $C^{\lambda_A,\lambda_B} > 0$ (for the cases of interest corresponding to $T > 0$). Thus, from (46), and since $b_{\{A\cap B\}} = 0$ for $\{A\cap B\} = \emptyset$,

$$b_A + b_B = b_{\{A\cup B\}} - C^{\lambda_A,\lambda_B} = b_{\{A\cup B\}} + b_{\{A\cap B\}} - C^{\lambda_A,\lambda_B}, \tag{47}$$

or,

$$b_A + b_B < b_{\{A\cup B\}} + b_{\{A\cap B\}}. \tag{48}$$

Case (II): Assume the A and B are not disjoint ($A\cap B \neq \emptyset$). By writing $\{A\cup B\}$ as the union of the disjoint sets $\{A\}$ and $\{B-\{A\cap B\}\}$ and using the results from Case I (46), the following can be written.

$$b_{\{A\cup B\}} = b_{\{A\cup\{B-\{A\cap B\}\}\}} = b_A + b_{\{B-\{A\cap B\}\}} + C^{\lambda_A,\lambda_{\{B-\{A\cap B\}\}}} \tag{49}$$

Similarly, by writing set B as the union of the disjoint sets $\{A\cap B\}$ and $\{B-\{A\cap B\}\}$, the following can be written,

$$b_B = b_{\{A\cap B\}} + b_{\{B-\{A\cap B\}\}} + C^{\lambda_{\{A\cap B\}},\lambda_{\{B-\{A\cap B\}\}}} \tag{50}$$

or,

$$b_{\{B-\{A\cap B\}\}} = b_B - b_{\{A\cap B\}} - C^{\dot\lambda_{\{A\cap B\}},\lambda_{\{B-\{A\cap B\}\}}}. \tag{51}$$

Finally, by combining (49) and (51), the following is obtained,

$$b_{\{A\cup B\}} + b_{\{A\cap B\}} = b_A + b_B + C^{\dot\lambda_A,\lambda_{\{B-\{A\cap B\}\}}} - C^{\lambda_{\{A\cap B\}},\dot\lambda_{\{B-\{A\cap B\}\}}} \tag{52}$$

If $A \subset B$ then,

$$C^{\dot\lambda_A,\lambda_{\{B-\{A\cap B\}\}}} - C^{\lambda_{\{A\cap B\}},\dot\lambda_{\{B-\{A\cap B\}\}}} = C^{\dot\lambda_A,\lambda_{\{B-A\}}} - C^{\dot\lambda_A,\lambda_{\{B-A\}}} = 0 \tag{53}$$

Similarly if $B \subset A$. If $A \not\subset B$, then $\{A \cap B\} \subset A$ and,

$$C^{\dot\lambda_A,\lambda_{\{B-\{A\cap B\}\}}} - C^{\lambda_{\{A\cap B\}},\dot\lambda_{\{B-\{A\cap B\}\}}} > 0, \tag{54}$$

completing the proof of the lemma. □

Lemma A.2. *The sets g_j, $j = 1, 2, \ldots, N$, that satisfy Eq. (20), defining the vertices of the polytope $\mathcal{D}^u$, must be strictly included in each other; that is*

$$\textit{either } h_j = g_j - (g_j \cap g_k) = \emptyset \textit{ or } h_k = g_k - (g_k \cap g_j) = \emptyset, \tag{55}$$

for all g_j and g_k in (20).

Proof. Let g_1 and g_2 be two sets satisfying (20). By adding the corresponding equations the following is obtained,

$$b_{\{g_1\}} + b_{\{g_2\}} = \sum_{i\in g_1} d_i^* + \sum_{i\in g_2} d_i^* = \sum_{i\in\{g_1\cup g_2\}} d_i^* + \sum_{i\in\{g_1\cap g_2\}} d_i^* \tag{56}$$

or,

$$\sum_{i\in\{g_1\cup g_2\}} d_i^* = b_{\{g_1\}} + b_{\{g_2\}} - \sum_{i\in\{g_1\cap g_2\}} d_i^*. \tag{57}$$

Since for any subset of S (17) must be satisfied, then for $\{g_1 \cap g_2\}$, $\sum_{i\in\{g_1\cap g_2\}} d_i^* \geq b_{\{g_1\cap g_2\}}$, the following is obtained from (57),

$$\sum_{i\in\{g_1\cup g_2\}} d_i^* \leq b_{\{g_1\}} + b_{\{g_2\}} - b_{\{g_1\cap g_2\}}. \tag{58}$$

Suppose now that g_1 and g_2 are not strictly included in each other. That is $h_1 = g_1 - (g_1 \cap g_2)$ and $h_2 = g_2 - (g_1 \cap g_2)$ are both non-empty. Then, Lemma A.1 implies that,

$$b_{\{g_1\}} + b_{\{g_2\}} < b_{\{g_1\cup g_2\}} + b_{\{g_1\cap g_2\}} \tag{59}$$

or,

$$b_{\{g_1\}} + b_{\{g_2\}} - b_{\{g_1\cap g_2\}} < b_{\{g_1\cup g_2\}}. \tag{60}$$

Therefore,

$$\sum_{i\in\{g_1\cup g_2\}} d_i^* \leq b_{\{g_1\}} + b_{\{g_2\}} - b_{\{g_1\cap g_2\}} < b_{\{g_1\cup g_2\}} \tag{61}$$

or,

$$\sum_{i\in\{g_1\cup g_2\}} d_i^* < b_{\{g_1\cup g_2\}}, \tag{62}$$

which implies $\mathbf{d}^* \notin \mathcal{D}^u$, this is a contradiction since $\mathbf{d}^*$ is a vertex of $\mathcal{D}^u$. Thus the assumption that g_i are strictly included in each other follows. □

References

[1] D. Bantz and F. Baucho, Wireless LAN Design Alternatives, *IEEE Network Magazine* 8(2) (1994) 43–45.

[2] P. Brady, A Model for On-Off Speech Patterns in Two-way Conversation, *Bell Systems Technical Journal* 48 (1969) 885–890.

[3] G. Chen and I. Stavrakakis, ATM Traffic Management with Diversified Loss and Delay Requirements, in: *Proceedings of IEEE INFOCOM*, San Francisco, CA (March 1996).

[4] E. Coffman and I. Mitrani, A Characterization of Waiting Time Performance Realizable by Single-Server Queues, *Operations Research* 28(3) (1980) 810–82.

[5] D. Goodman, R. Valenzeula et al., Packet Reservation Multiple Access for Local Wireless Communication, *IEEE Transactions on Communications* 37(8) (1989) 885–890.

[6] M. Karol, Z. Liu and K. Eng, An Efficient Demand-Assignment Multiple Access Protocol for Wireless Packet (ATM) Networks, *Wireless Networks* 1(4) (1995) 267–279.

[7] F. Li and L. Merakos, Voice Data Integration in Digital TDMA Cellular System, in: *Proceedings of International Zurich Seminar on Mobile Communications*, Zurich, Switzerland (March 1994).

[8] S. Nanda, D. Goodman and U. Timor, Performance of PRMA: A Packet Voice Protocol for Cellular Systems, *IEEE Transactions on Vehicular Technology* 40(3) (1991) 584–589.

[9] A. Brøndsted, *An Introduction to Convex Polytopes*, Springer-Verlag (1983).

[10] D. Raychaudhuri and N. Wilson, ATM-Based Transport Architecture for Multiservices Wireless Personal Communication Networks, *IEEE Journal of Selected Areas Communications* 12(8) (1994) 1401–1414.

[11] K.W. Ross and D.D. Yao, Optimal Dynamic Scheduling in Jackson Networks, *IEEE Transactions on Automatic Control* 34(1) (1989) 47–53.

[12] W. Sharkey, Cooperative Games with Large Cores, *International Journal of Game Theory* 11 (1982) 175–182.

[13] D. Welsh, *Matroid Theory*, Academic Press (1976).

Study of Delay Jitter With and Without Peak Rate Enforcement

Randall Landry, *Member, IEEE*, and Ioannis Stavrakakis, *Senior Member, IEEE*

Abstract—**In this work, the modification of a tagged traffic stream due to statistical multiplexing is studied by presenting a numerical approach for the calculation of the tagged delay jitter and interdeparture processes. Both the single- and multiple-node cases are considered. Unlike the past work, the developed approach is applicable under both the standard FCFS policy and a peak-rate enforcing multiplexing policy. The latter policy can be adopted to reshape the tagged traffic stream within the network. Restoring a shaped traffic profile within the network may be necessary to obtain some of the benefits for which the original shaping is carried out at the network edge. This study also provides results and insight regarding the potential gains of reregulation within the network.**

Index Terms— **author: please supply.**

I. Introduction

The asynchronous transfer mode (ATM) is presently considered to be one of the most promising switching and multiplexing schemes for B-ISDN's. Although it is possible to achieve high levels of network efficiency (utilization) through the statistical multiplexing of bursty sources, it is not always easy to guarantee quality of service (QoS). Since at a given time instant the cumulative peak rate of the supported ATM users may exceed the network capacity, serious congestion may occur, resulting in QoS degradation. Source traffic shaping and/or regulation has been proposed as a means of reducing traffic burstiness, thereby controlling network congestion.

It is well understood that statistical multiplexing may induce substantial distortion in an ATM cell stream [1]. Thus, characterization of a tagged traffic stream within the network seems to be necessary in order to accurately evaluate potential congestion, as well as determine the QoS delivered to the tagged stream through a per-session end-to-end study. In addition, a corrective action (that is, traffic shaping) may be necessary within the network for the purpose of reducing distortion, which is a potential cause of network congestion.

In this work, the distortion of a (regulated) source traffic stream due to a series of multiplexing operations is studied under the standard FCFS, as well as a "corrective" peak-rate enforcing, multiplexing discipline. The tagged traffic stream is characterized at the output of a multiplexing node in terms of the cell delay, delay jitter, and interdeparture processes. A comparative study of the two multiplexing disciplines is carried out based on these quantities, and useful insight is obtained regarding the resulting distortion of the tagged traffic stream. Finally, the node-by-node approximate description of the transformation of a tagged stream under the considered multiplexing disciplines is used for an approximate study of the end-to-end QoS delivered to the tagged traffic stream.

Past work regarding the characterization of a tagged traffic stream due to a single FCFS multiplexing operation may be found in [2], [3]. In [2], a method is developed for calculating the cell interdeparture time distribution of a general renewal stream under an FCFS policy. Statistics for the interdeparture process of an initially periodic cell stream in a FCFS multiplexer are also computed in [3] by assuming a first-order Markovian structure for the cell delay process.

The problem of measuring end-to-end performance in an ATM environment is difficult and largely open. Typically, a specific application (session) is being tagged and observed as it traverses the network. Deterministic bounds on delay have been derived in [4]–[7], where the traffic from each session is assumed to conform to predefined burstiness constraints. These approaches are based on worst case performance and the bounds are extremely loose, resulting in the underutilization of available resources when used for call admission purposes [8]. Provable statistical bounds have been derived in [9], [10], which also appear to be too loose to be of practical use in call admission. An alternative approach is to provide approximate statistical QoS guarantees [11]–[13] by modeling the traffic at the network edge (or within the network itself) and analyzing the resulting queueing systems. While this approach has the advantage of simplicity, it generally suffers from the traffic assumptions which must be made at the network premises. Past work on approximate end-to-end network performance evaluation based on nodal decomposition and FCFS multiplexing may be found in [14]–[17]. Although the end-to-end study presented in this paper is based on "standard" nodal decomposition assumptions as in [14], [17], the single-node analysis approach is different, allowing for the consideration of policies which cannot be studied by employing the past approaches.

In the next section, the fundamental block of this work is considered, namely, a queueing system under a service policy which occasionally departs from the FCFS policy to enforce a peak output rate (minimum cell interdeparture time) for the tagged traffic stream (Class-1). It is shown (Section II-C) that

Manuscript received November 9, 1994; revised February 27, 1996 and January 28, 1997; approved by IEEE/ACM Transactions on Networking Editor D. Mitra. This work was supported by the Advanced Research Projects Agency under Grant F49620-93-1-0564 monitored by the Air Force Office of Scientific Research (AFOSR).

R. Landry is with the DSP R&D Center, Texas Instruments Inc. (e-mail: landry@ti.com).

I. Stavrakakis is with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115 USA.

Publisher Item Identifier S 1063-6692(97)05576-3.

1063–6692/97$10.00 © 1997 IEEE

the proposed service policy allows for a characterization of the tagged cell stream which is consistent with the GCRA defined in [18]. Furthermore, when the peak-rate enforcer is disabled (by properly setting a key parameter), the proposed policy becomes the standard FCFS policy considered in [19], [3]. In Section IV, the approximate end-to-end study is described along with a discussion of important assumptions. Numerical results are presented in Section IV before concluding in the final section.

II. ANALYSIS OF A SINGLE MULTIPLEXING STAGE

There are two main objectives in this work. First, the distortion in a tagged traffic stream at a single node due to the standard FCFS as well as a distortion-reducing service policy will be studied. This study will reveal the amount of distortion introduced by the policies considered, as well as provide for a description of the tagged traffic stream at the output of the multiplexing stage. The latter corresponds to the tagged stream arriving at the next multiplexing stage, and thus, the next stage can be studied in a similar manner considering the tagged traffic stream as shaped by the previous multiplexing stage. By reiterating the above over a series of multiplexing stages, an approximate end-to-end study of a tagged traffic stream can be carried out, revealing the end-to-end impact of the FCFS and the distortion-reducing multiplexing disciplines on a tagged stream, which is the second major objective of this work.

A. The Distortion-Reducing Peak Output-Rate Enforcing (PORE) Service Discipline

The PORE service discipline is associated with a tagged traffic stream (Class-1), and treats the rest of the traffic in an indistinguishable manner (Class-2 or background traffic). As it becomes clear below, the PORE service strategy for Class-1 guarantees that cells belonging to this class are transmitted at a minimum spacing of $X_{\min}$ slots. The server operates according to the FCFS discipline, except for those times when the FCFS server would place consecutive Class-1 cells on the output link at a distance of less than $X_{\min}$. In this paper, it is assumed that under the FCFS policy, Class-1 cells are served before the Class-2 cells which arrive over the same slot.

Let t_k denote the service epoch of the kth Class-1 cell, defined as the beginning of the slot over which this cell is transmitted. Let $X_k = t_{k+1} - t_k$ denote the interdeparture time of Class-1 cells $k+1$ and k. Similarly, $\tilde{t}_k$ will mark the *potential* service epoch of the kth Class-1 cell, which is defined to be the beginning of the slot at which all other cells that arrived prior to the kth Class-1 cell have been served. Note that under the FCFS discipline, $\tilde{t}_k = t_k, \forall k \geq 1$.

The PORE service discipline can be described in terms of the following rules.

1) Whenever $\tilde{t}_k - t_{k-1} < X_{\min}$, the server delays the initiation of service to the kth Class-1 cell by exactly $H_k = X_{\min} - (\tilde{t}_k - t_{k-1})$ slots, which will be referred to as the *holding time* of the kth cell for the remainder of the paper.
2) During this holding time of length H_k, the server will provide FCFS service to Class-2 cells that may be present in the buffer. This implies that during the holding interval, up to H_k Class-2 cells can be served before the kth Class-1 cell which arrived at an earlier time.
3) When the holding time expires, the kth Class-1 cell is served; thus, $t_k = \tilde{t}_k + H_k$, thereby ensuring that $X_{k-1} = X_{\min}$.

Note that according to rule 2), the server abandons the FCFS discipline whenever a Class-1 cell is held ($H_k \geq 1$), and thus it provides improved service to Class-2 compared to that under the FCFS policy. The PORE service policy reduces to the FCFS policy when $X_{\min} = 1$, in which case $H_k = 0, \forall k \geq 1$.

Let the Class-1 cell interarrival process be denoted by $\{A_k\}_{k\geq 1}$, where A_k represents the interarrival time (in slots) between Class-1 cells k and $k+1$. It is assumed that $A_k \in [A_{\min}, \infty) \subseteq Z^+$, where Z^+ denotes the set of positive integers. Class-2 traffic is considered to be the background traffic whose presence stands to corrupt $\{A_k\}_{k\geq 1}$ through the queueing process.

B. Jitter (Cell Delay Variation) and the PORE Policy

An important measure of distortion of a real-time application's cell stream is delay jitter, or cell delay variation. When a session experiences a large amount of delay jitter, or distortion to its traffic profile, extensive buffering may be required at the receiver in order for the original data stream to be recreated before being played back. Clearly, this implies that network-induced delay jitter can be removed at the receiver at the expense of larger buffers and potentially increased delays. In view of this, it seems reasonable to expect that the end user buffering requirements can potentially be significantly reduced by controlling delay jitter within the network. The implication here is that some of the buffers can be displaced from the receiving end to the network premises where resources are shared by more than one user. Finally, besides creating the need for increased buffering at the receiver, an initially well-behaved cell stream can induce queueing problems and QoS degradation for existing sessions as it becomes distorted and potentially more bursty within the network.

A number of definitions for delay jitter have appeared in the literature, including those which compute the probability distribution of cell delays and measure the difference between lower and upper quantiles [3], [20]. In this paper, a cell-level definition for delay jitter, like the one considered in [3], will be adopted to describe the amount of distortion introduced into the Class-1 cell stream.

Definition 1: Let D_k denote the total queueing delay (in slots) of the kth Class-1 cell, including the transmission slot. The delay jitter of the kth cell is defined as

$$J_k = D_{k+1} - D_k \qquad \text{for } k \geq 1. \quad \square \tag{1}$$

In terms of the quantities defined in the previous subsection, an equivalent definition of delay jitter is given by

$$J_k = X_k - A_k \qquad \text{for } k \geq 1. \tag{2}$$

Note that J_k measures the degree to which the interarrival time A_k of Class-1 cells k and $k+1$ has been distorted, and it can take both positive and negative values. *Negative* delay jitter applies to those instances when Class-1 cells k and $k+1$ are transmitted at a spacing of less than A_k (cell clustering), while *positive* delay jitter applies to those times when cells k and $k+1$ are transmitted at a spacing of greater than A_k (cell spreading). Note that when the process $\{A_k\}_{k\geq 1}$ has a finite region of support (i.e., $A_k \in [A_{\min}, A_{\max}]$), the PORE policy completely eliminates a portion of the negative delay jitter by guaranteeing that $J_k \geq X_{\min} - A_{\max}$.

Since providing a minimum spacing of $X_{\min}$ eliminates some of the negative delay jitter, it would be reasonable to investigate whether the PORE policy induces probabilistically larger positive delay jitter than the FCFS policy. This is addressed in the following theorem.

Theorem 1: Consider any Class-1 arrival stream with minimum cell interarrival time $A_{\min}$. Let $J_k^{\rm FCFS}$ ($J_k^{\rm PORE}$) denote the delay jitter of the kth Class-1 cell under the FCFS (PORE) service policy. If $X_{\min} < A_{\min}$, then for all $k \geq 2$,

$$\Pr\{J_k^{\rm PORE} \geq j\} \leq \Pr\{J_k^{\rm FCFS} \geq j\} \qquad \forall j \geq 0$$

with equality when $X_{\min} = 1$.

Proof: Recall that if the kth Class-1 cell is held, it is guaranteed to be served at time t_k such that $X_k = X_{\min}$. This implies that a held cell will never be placed on the output link at a distance less than $X_{\min}$ from the previous Class-1 cell. Since $X_{\min}$ is always less than $A_{\min}$, then $X_k < A_k$. From (2), this implies that $J_k = X_k - A_k$ cannot be greater than or equal to zero, or positive delay jitter is never induced through this output-rate-enforcement action. In fact, the positive delay jitter is decreased since, while the kth Class-1 cell is being held, the server is attending to Class-2, and consequently, the service time of the $(k+1)$st cell is possibly moving closer to t_k. □

Theorem 1 establishes the fact that regulating Class-1 by enforcing a peak output rate does not increase the positive delay jitter of this traffic class. The PORE service policy actually reduces the two types of distortion most commonly induced by multiplexers, namely, spreading and clustering.

C. The Generic Cell Rate Algorithm (GCRA) for ATM

The negative impact of delay jitter on an ATM connection (as well as other supported sessions in the network) has been recognized by the ATM Forum [18], and one of the consequences has been the development of the GCRA, which establishes a relationship between peak cell rate R_p and cell delay variation tolerance (CDVT). R_p is defined as the inverse of the minimum cell interarrival time at the user; assuming that $A_{\min}$ is the minimum interarrival time contracted by the user, then $R_p = 1/A_{\min}$. On the other hand, CDVT represents a bound on the cell clustering phenomenon previously described, and it is defined in relation to R_p according to the GCRA as explained below.

The basic function of the GCRA is to update a theoretical arrival time (TAT), which is simply the expected arrival time of a cell assuming that cells are equispaced when the source is active. Let t_k^a denote the actual arrival time of the kth cell at some observation point within the network. If t_k^a is not "too early" relative to the TAT, or if $t_k^a \geq$ TAT $- \tau$ for some small $\tau > 0$, then the cell is said to be *conforming*. The corresponding cell stream is said to conform to GCRA($A_{\min}$, τ), where τ is the value of CDVT. In this paper, it will be assumed that τ takes only integer values.

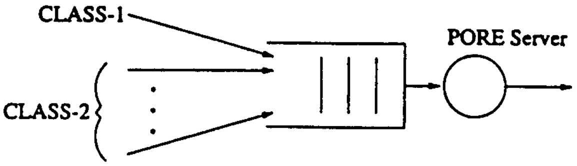


Fig. 1. The discrete-time queueing system.

A cell stream conforming to GCRA($A_{\min}$, τ) may be considered to have a contracted peak rate of $1/A_{\min}$ and a "tolerated" peak rate of $1/X_{\min}$, where

$$X_{\min} = A_{\min} - \tau.$$

Clearly, the PORE service policy described above can be used to ensure that the Class-1 cell stream conforms to GCRA($A_{\min}$, τ) at the output of any multiplexer or switching site within the network.

The appropriate value of τ, which is to be chosen by the user from a set of values supported by the network, is the topic of ongoing research [18], but it is expected that under certain operating conditions, the impact of CDVT on network resources will be significant. Clearly, a τ decreases the more well behaved the cell stream becomes. In terms of the quantities associated with the PORE policy, when $X_{\min} = A_{\min}$, τ becomes zero and the Class-1 cell stream conforms to GCRA($A_{\min}$, 0), which happens to be the function of the Virtual Shaper defined in [18]. Providing this strict conformance with the contracted peak rate will undoubtedly have a positive impact on the QoS delivered to other traffic streams along the same Virtual Path. As previously mentioned, the present study will focus on the impact that CDVT has on the conforming (or tagged) cell stream in terms of delay jitter, cell interarrival times, and cell delays.

D. Queueing Analysis

In this section, an ATM multiplexer operating under the PORE service policy (Fig. 1) is studied in terms of performance measures associated with the Class-1 cell stream. The Class-1 interarrival process $\{A_k\}_{k\geq 1}$ is assumed to be i.i.d. with distribution $f_a(m)$, $A_{\min} \leq m \leq A_{\max}$. Note that a periodic process with period T can be considered when $A_{\min} = A_{\max} = T$ and $f_a(T) = 1$.

Class-2 traffic is modeled in terms of i.i.d. batches of cells of arbitrary distribution. Let the random variable B_j—with probability mass function (PMF) $f_{b_j}(k)$, $k \geq 0$—denote the number of Class-2 cells arriving over an interval of length j (in slots). Clearly, $f_{b_j}(\cdot)$ can be derived as the j-fold convolution of $f_{b_1}(\cdot)$; let N^b be the maximum value of B_1.

Let $Q(n; j)$ denote the number of Class-2 cells which arrived before time epoch j and are still in the system at time n, $n \geq j$. The following proposition determines the evolution of $\{Q(n; j)\}_{n\geq j}$ under certain conditions, and will be useful for the study of the queueing system.

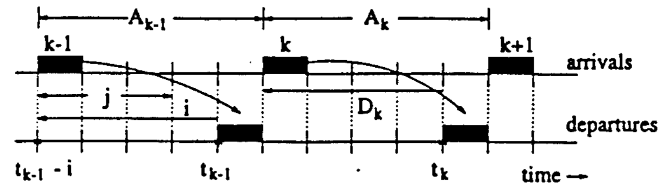


Fig. 2. Realization for the derivation of D_k and H_k when $i < A_{\max}$.

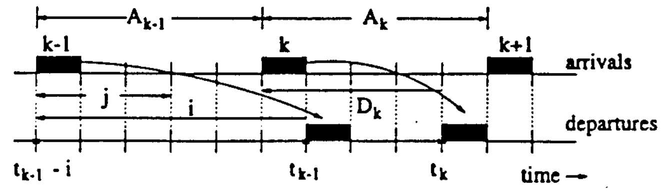


Fig. 3. Realization for the derivation of D_k and H_k when $i \geq A_{\max}$.

Proposition 1: The process $\{Q(j+m;\, j)\}_{m=0}^{n}$ evolves as the occupancy process of a Geo/D/1 queue with batch arrivals distributed according to $f_{b_1}(\cdot)$, provided that *no* Class-1 cells are served over the interval $[j,\, j+n]$. In this case,

$$\begin{aligned}\Pr\{Q(j+m;\, j) = \ell \mid Q(j;\, j) = k\} &= P_{k,\ell}^{(m)} \\ &= \sum_{i} P_{k,i}^{(m-1)} P_{i,\ell} \qquad (3)\end{aligned}$$

where $P_{k,\ell}^{(m)}$ $(P_{i,\ell})$ is the m-step (one-step) transition probability for the Geo/D/1 queue; $P_{i,\ell} = f_{b_1}(\ell - i + 1) + f_{b_1}(0)1_{\{\ell+i=0\}}$. □

The queueing system will be described in terms of the Markov chain $\{D_k,\, H_k\}_{k\geq 1}$, embedded at the sequence of time instants $\{t_k\}_{k\geq 1}$; t_k marks the service epoch of the kth Class-1 cell as defined earlier; D_k and H_k denote the total delay and holding time of the kth cell, respectively.

The transition probabilities for the Markov chain $\{D_k,\, H_k\}_{k\geq 1}$ will be derived by first establishing expressions for D_k and H_k given that $(D_{k-1},\, H_{k-1}) = (i,\, j)$, where $i \geq 1$ and $0 \leq j \leq X_{\min} - 1$. In order to simplify notation, the quantity $Q(t_{k-1};\, t_{k-1} - i + j)$, which appears frequently in the following expressions, will be denoted by Q^*. Also, let Q_k denote the number of Class-2 cells in queue at the arrival epoch (time $t_{k-1} - i + A_{k-1}$) of the kth Class-1 cell. That is, $Q_k = Q(t_{k-1} - i + A_{k-1};\, t_{k-1} - i + A_{k-1})$.

When $i < A_{\max}$, the expressions for D_k and H_k can be derived by referring to Fig. 2, where $i < A_{k-1}$ for the realization shown there. Note that i can also take values greater than or equal to A_{k-1} while still being less than $A_{\max}$, and these cases are also accounted for in the expressions given below in (4) and (5), when $i \geq A_{\max}$ (Fig. 3), shown in (6) and (7).

Let $S^L = \{(i,\, j) : L(A_{\max} - 1) + 1 \leq i \leq (L+1)(A_{\max} - 1),\, 0 \leq j \leq X_{\min} - 1\}$ represent "level" L of the Markov chain $\{D_k,\, H_k\}_{k\geq 1}$. Given the relationships in (4)–(7), the transition probabilities for $\{D_k,\, H_k\}_{k\geq 1}$, which will be denoted by $p(i,\, j,\, i',\, j')$, can be computed in terms of the quantities previously defined. The quantities $p(i,\, j,\, i',\, j')$, for $(i,\, j) \in S^0$ (transitions from level 0) are given by (8) and (9), which are derived from (4) and (5) above, while the transition probabilities for $(i,\, j) \in S^1$ are given by (10) and (11), derived from (6) and (7) above.

For $(i,\, j) \in S^0$,

$$\begin{aligned}p(i,\, j,\, i',\, 0) = &\sum_{n=A_{\min}}^{A_1} \sum_{m=0}^{N_1} f_a(n) b_{n-j}(m) P_{0,\, i'-i+n-m-1}^{(j)} \\ &+ \sum_{n=A_2}^{A_{\max}} \sum_{l=0}^{(N^b-1)j} \sum_{m=0}^{N^b(i-j+1)} f_a(n) P_{0,l}^{(j)} \\ &\cdot b_{i-j+1}(m)[P_{m+l,\, i'}^{(n-i-1)} + P_{m+l,\, 0}^{(n-i-1)} 1_{\{i'=1\}}], \qquad (8)\end{aligned}$$

where $A_1 = \min\{i,\, A_{\max}\}$, $A_2 = \max\{A_{\min},\, i+1\}$ and $N_1 = \min\{(n-j)N^b,\, i' - 1 - i + n\}$.

$$\begin{aligned}p(i,\, j,\, i',\, j') = &\sum_{n=A_{\min}}^{A_1} \sum_{m=0}^{N_2} f_a(n) b_{n-j}(m) P_{0,\, X_{\min}-1-m-j'}^{(j)} \\ &+ \sum_{n=A_2}^{A_{\max}} \sum_{l=0}^{(N^b-1)j} \sum_{m=0}^{N^b(i-j+1)} f_a(n) P_{0,l}^{(j)} \\ &\cdot b_{i-j+1}(m)[P_{m+l,\, i'-j'}^{(n-i-1)} + P_{m+l,\, 0}^{(n-i-1)}] \\ &\cdot 1_{\{i'=X_{\min}-n+i\}} \qquad (9)\end{aligned}$$

where $N_2 = \min\{(n-j)N^b,\, X_{\min} - 1 - j'\}$.

For $(i,\, j) \in S^1$,

$$\begin{aligned}p(i,\, j,\, i',\, 0) = &\sum_{n=A_{\min}}^{A_1} \sum_{m=0}^{N_1} f_a(n) \\ &\cdot b_{n-j}(m) P_{0,\, i'-i+n-m-1}^{(j)} \qquad (10)\end{aligned}$$

and

$$\begin{aligned}p(i,\, j,\, i',\, j') = &\sum_{n=A_{\min}}^{A_1} \sum_{m=0}^{N_2} f_a(n) b_{n-j}(m) \\ &\cdot P_{0,\, X_{\min}-1-m-j'}^{(j)} 1_{\{i'=X_{\min}-n+i\}}. \qquad (11)\end{aligned}$$

By lexicographically mapping the quantities $p(i,\, j,\, i',\, j')$ into $p(\ell,\, \ell')$ through the relations $\ell = (i-1)X_{\min} + j$ and $\ell' = (i'-1)X_{\min} + j'$, the transition probability matrix $\mathbf{P}$ of the Markov chain $\{D_k,\, H_k\}_{k\geq 1}$ can be written as a stochastic matrix of the M/G/1 type:

$$\mathbf{P} = \begin{bmatrix} \mathbf{B}_0 & \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{B}_3 & \cdots \\ \mathbf{A}_0 & \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3 & \cdots \\ 0 & \mathbf{A}_0 & \mathbf{A}_1 & \mathbf{A}_2 & \cdots \\ 0 & 0 & \mathbf{A}_0 & \mathbf{A}_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \qquad (12)$$

where the matrices $\mathbf{B}_n$ and $\mathbf{A}_n$, $n \geq 0$, have dimensions $(A_{\max}-1)X_{\min} \times (A_{\max}-1)X_{\min}$. Note that the matrices $\mathbf{B}_n$ consist of the transition probabilities for which $(i,\, j) \in S^0$ and $(i',\, j') \in S^n$, while the matrices $\mathbf{A}_n$ consist of the transition probabilities for which $(i,\, j) \in S^1$ and $(i',\, j') \in S^n$. The system in (12) can now be solved as outlined in the Appendix based on the matrix analytic techniques presented in [21]. The stationary probability vector $\boldsymbol{\pi}$ for $\mathbf{P}$ is mapped back into the space (i, j, i', j') to obtain the stationary probabilities for $\{D_k, H_k\}_{k\geq 1}$, given by $\pi(i, j)$, for $1 \leq i < \infty$, $0 \leq j \leq X_{\min} - 1$.

The PMF for cell delay D of Class-1 is easy to calculate in terms of $\pi(i, j)$, and it is given by

$$\Pr\{D = i\} = \sum_{j=0}^{X_{\min}-1} \pi(i, j). \tag{13}$$

By considering the definition in (1), the delay jitter distribution is easily determined to be

$$\Pr\{J = k\} = \sum_{i=1}^{\infty} \sum_{j=0}^{X_{\min}-1} \sum_{j'=0}^{X_{\min}-1} p(i, j, i+k, j')\pi(i, j). \tag{14}$$

Computing the PMF of the Class-1 cell interdeparture time X is not as straightforward as computing that of J. From (1) and (2), the interdeparture time of cells $k+1$ and k is given as $X_k = D_{k+1} - D_k + A_k$. Since the Markov chain $\{D_k, H_k\}_{k\geq 1}$ does not provide adequate information to describe the interdeparture process, the system will be described by the Markov chain $\{D_k, H_k, A_k\}_{k\geq 1}$ in order to compute the PMF of X. Let $\bar{p}(i, j, m, i', j', m')$ denote the transition probabilities, and $\bar{\pi}(i, j, m)$ denote the stationary probabilities for this Markov chain. It is easy to show that the PMF for the cell interdeparture process is given by

$$\Pr\{X = n\} = \sum_{i=1}^{\infty} \sum_{j=0}^{X_{\min}-1} \sum_{m=A_{\min}}^{A'} \sum_{j'=0}^{X_{\min}-1} \sum_{m'=A_{\min}}^{A_{\max}} \cdot \bar{p}(i, j, m, n+i-m, j', m')\bar{\pi}(i, j, m) \tag{15}$$

where $A' = \min\{n + i - 1, A_{\max}\}$.

The quantities $\bar{\pi}(i, j, m)$ can be computed in the same manner as the quantities $\pi(i, j)$, with the exception that the dimensionality of each level in $\mathbf{P}$ has increased by a factor of $A_{\max}$. Note, however, that upon arrival of the kth Class-1 cell, D_k is determined. That is, D_k is equal to the queue content at the arrival instant plus the amount of holding time H_k to be suffered by the cell. The latter quantity is completely determined by the past (i.e., the service time of the $(k-1)$st cell), and neither D_k nor H_k is dependent upon the next Class-1 arrival. Consequently, the random variables D_k and H_k are both independent of the random variable A_k, and the stationary probabilities for the Markov chain $\{D_k, H_k, A_k\}_{k\geq 1}$ can be computed as $\bar{\pi}(i, j, m) = \pi(i, j)f_a(m)$, where $\pi(i, j)$ and $f_a(m)$ are as given above.

III. Approximate End-to-End Study

The N-node tandem queueing system used for the approximate end-to-end performance evaluation of the tagged session is shown in Fig. 4. The PORE service policy will be adopted at each node so as to allow for the consideration of peak-rate enforcement within the network. In order to simplify notation, the Class-1 cell interarrival (interdeparture) process at node i will be denoted as A^i (X^i). The arrival process at node i $(1 \leq i \leq N)$ consists of the tagged (Class-1) interdeparture process from node $i-1$, and some background traffic (Class-2) delivering batches of cells distributed according to the PMF $f_{b_1}(\cdot)$. X^0 is defined to be the tagged cell intergeneration distance at the source; A^1 may be slightly different from X^0 due to potential truncation (as discussed below).

The set of possible integer values which can be taken by the Class-1 interarrival processes at nodes 1–N must be truncated in order to facilitate a numerical solution at each node. The minimum (maximum) Class-1 interarrival time at node i will be denoted by $A^i_{\min}$ $(A^i_{\max})$. That is, $A^i \in [A^i_{\min}, A^i_{\max}]$ for $i = 1, 2, \cdots, N$. The minimum interarrival time for $i \geq 1$ is simply $A^i_{\min} = X^{i-1}_{\min}$. The choice of $A^i_{\max}$ for $i \geq 1$ will be

$$D_k = \begin{cases} 1 & \text{if } Q_k = 0 \text{ and } A_{k-1} \geq X_{\min} + i - 1 \text{ and } i < A_{k-1} \\ Q_k & \text{if } Q_k \geq X_{\min} + i - A_{k-1} \text{ and } i < A_{k-1} \\ Q_k + H_k & \text{if } Q_k < X_{\min} + i - A_{k-1} \text{ and } i < A_{k-1} \\ B_{A_{k-1}-j} + Q^* + 1 + i - A_{k-1} & \text{if } Q^* \geq X_{\min} - B_{A_{k-1}-j} - 1 \text{ and } i \geq A_{k-1} \\ B_{A_{k-1}-j} + Q^* + 1 + i - A_{k-1} + H_k & \text{if } Q^* < X_{\min} - B_{A_{k-1}-j} - 1 \text{ and } i \geq A_{k-1} \end{cases} \tag{4}$$

and

$$H_k = \begin{cases} 0 & \text{if } Q_k = 0 \text{ and } A_{k-1} \geq X_{\min} + i - 1 \text{ and } i < A_{k-1} \\ 0 & \text{if } Q_k \geq X_{\min} + i - A_{k-1} \text{ and } i < A_{k-1} \\ X_{\min} + i - A_{k-1} - Q_k & \text{if } Q_k < X_{\min} + i - A_{k-1} \text{ and } i < A_{k-1} \\ 0 & \text{if } Q^* \geq X_{\min} - B_{A_{k-1}-j} - 1 \text{ and } i \geq A_{k-1} \\ X_{\min} - B_{A_{k-1}-j} - Q^* - 1 & \text{if } Q^* < X_{\min} - B_{A_{k-1}-j} - 1 \text{ and } i \geq A_{k-1}. \end{cases} \tag{5}$$

$$D_k = \begin{cases} B_{A_{k-1}-j} + Q^* + 1 + i - A_{k-1} & \text{if } Q^* \geq X_{\min} - B_{A_{k-1}-j} - 1 \text{ and ¿}i \geq A_{k-1} \\ B_{A_{k-1}-j} + Q^* + 1 + i - A_{k-1} + H_k & \text{if } Q^* < X_{\min} - B_{A_{k-1}-j} - 1 \text{ and } i \geq A_{k-1} \end{cases} \tag{6}$$

and

$$H_k = \begin{cases} 0 & \text{if } Q^* \geq X_{\min} - B_{A_{k-1}-j} - 1 \text{ and } i \geq A_{k-1} \\ X_{\min} - B_{A_{k-1}-j} - Q^* - 1 & \text{if } Q^* < X_{\min} - B_{A_{k-1}-j} - 1 \text{ and } i \geq A_{k-1}. \end{cases} \tag{7}$$

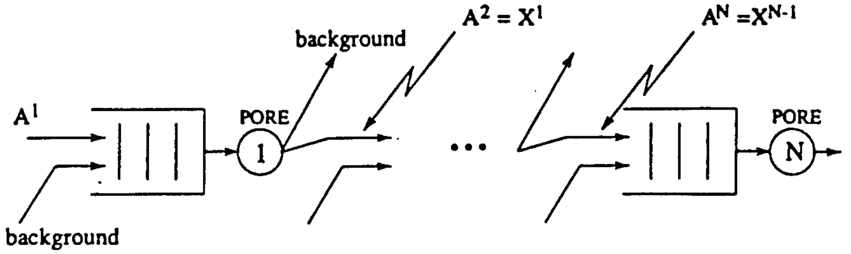


Fig. 4. Tandem queueing network for end-to-end performance evaluation.

made according to the rule

$$A^i_{\max} = \max\{k\colon \Pr\{X^{i-1} = k\} \geq \epsilon\} \qquad (16)$$

where ϵ can be any positive real number less than 1. The PMF for the process A^i, $i \geq 1$, is then given by

$$\Pr\{A^i = k\} = \begin{cases} \Pr\{X^{i-1} = k\} & \text{if } 0 < k < A^i_{\max} \\ 1 - \sum_{j=1}^{A^i_{\max}-1} \Pr\{X^{i-1} = j\} & \text{if } k = A^i_{\max} \\ 0 & \text{otherwise.} \end{cases} \qquad (17)$$

It is clear from (16) and (17) that as ϵ approaches zero, $A^i_{\max}$ increases and the truncated Class-1 interarrival time PMF at a node approaches the actual Class-1 interdeparture time PMF from the previous node. The cost of improving the accuracy of the adopted Class-1 interarrival process by increasing $A^i_{\max}$ is a corresponding increase in numerical complexity. Recall that the dimensionality of the square matrices B_n and A_n, $n \geq 0$, which make up the transition probability matrix $\mathbf{P}$ for a single queue, increases linearly with the maximum Class-1 interarrival time (see Section II-D), which in turn affects the tractability of the solution technique outlined in the Appendix.

A number of other assumptions will be made in order to facilitate the proposed approximate end-to-end study. First, it will be assumed that only the tagged traffic stream is forwarded from one multiplexer to the next as shown in Fig. 4. All of the remaining traffic (background) at each multiplexer is routed elsewhere. Implicit in this assumption is the presence of a large-scale networking structure, in which there exist a large number of independent sessions and a large number of total nodes (only N of which serve the tagged session). In such a network, it is expected to be reasonable to assume that, at the output of each node, most of the untagged sessions (background traffic) are split, or routed along paths which are different from that of the tagged session.

It is also assumed that the interdeparture process $\{X_k\}_{k\geq 1}$ from each node is an i.i.d. process. This assumption is necessary for employment of the analysis of Section II-B at each multiplexing stage. The validity of the i.i.d. assumption, as well as the effect of truncating the interarrival time PMF at each node, will be examined through a comparative simulation study.

The analysis presented in the previous section for a single queue will be applied successively to each node along the path in Fig. 4, and the PMF for the interdeparture process at node N will be used to establish the amount of delay jitter induced in the cell stream of the tagged session over all N hops.

TABLE I
SQUARED COEFFICIENT OF VARIATION FOR CELL INTER-DEPARTURE TIME (C_x^2)

load	$T = 5$	$T = 10$	$T = 20$
0.70	0.0236	0.0086	0.0028
0.80	0.0489	0.0209	0.0079
0.95	0.1160	0.0607	0.0296

IV. NUMERICAL RESULTS AND DISCUSSION

In this section, some numerical results are presented to characterize the distortion induced in a tagged traffic stream under both FCFS and peak-rate enforcing service strategies. Before considering end-to-end performance, the delay jitter induced at a single ATM multiplexer will be studied.

In order to provide for a simple interpretation of the multiplexer-induced distortion, the Class-1 arrival process is assumed to be periodic with period T slots in the results presented in this section. In addition, a typical regulated traffic stream (whose reshaping may be attempted within the network) can be considered to resemble a periodic one for some time horizon which—in some cases—may be sufficient to induce "local" stationary statistics. It should be emphasized, though, that the study presented in this paper is applicable to traffic streams described in terms of general interarrival processes. The Class-2 batch arrival process is assumed to be binomially distributed with mean rate λ^b and maximum batch size $N^b = 15$. Table I displays the squared coefficient of variation for the Class-1 cell interdeparture time process when an FCFS policy is adopted. The squared coefficient of variation will be denoted by C_X^2 and given as

$$C_x^2 = \frac{\operatorname{var}(X)}{E^2(X)}.$$

Note that C_x^2 is equal to zero if no distortion is introduced into the periodic arrival stream. From Table I, it is clear that delay jitter is the greatest, under the FCFS policy, for small periods T and high offered loads.

The multiplexer-induced distortion can also be quantified by observing the PMF for J as in Fig. 5. In this case, the delay jitter PMF is plotted for $T = 20$ and the offered loads given in Table I. Notice that when the total load is 0.7, much of the periodicity is retained in the Class-1 arrival stream, which is illustrated by the fact that most of the probability mass exists at $J = 0$. However, as the load increases to 0.95, much of this mass is redistributed and the Class-1 arrival process no longer resembles a periodic source. It is also interesting to

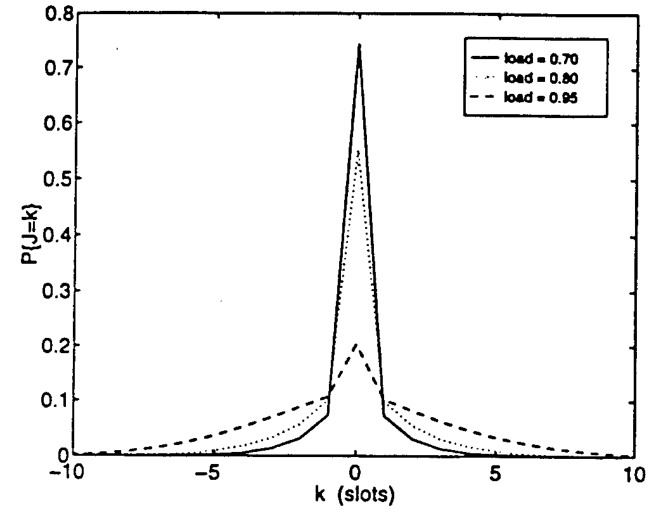


Fig. 5. Jitter PMF at an FCFS multiplexer for $T = 20$ and various loads.

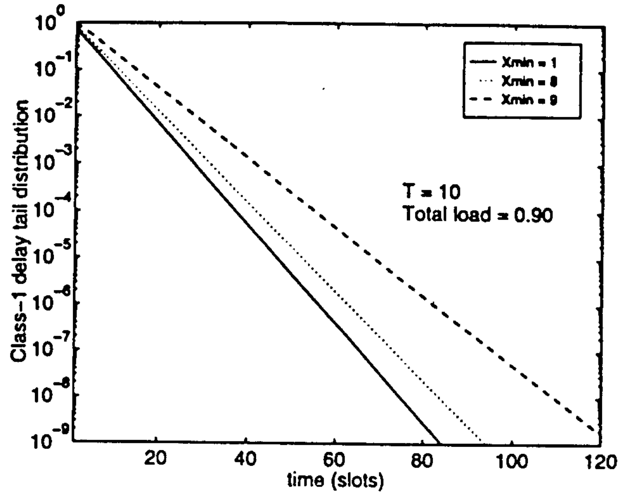


Fig. 7. Delay tail distributions for $T = 10$ and various values of $X_{\min}$ at a single multiplexer.

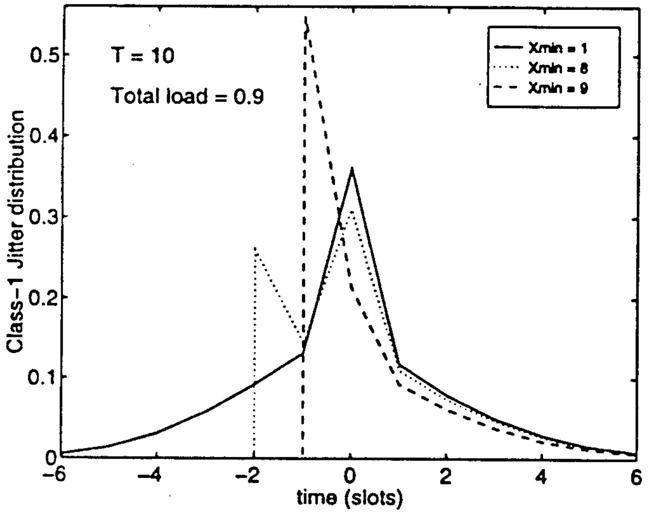


Fig. 6. Jitter PMF for $T = 10$ and various values of $X_{\min}$ at a single multiplexer.

note that the delay jitter PMF appears to be symmetric around zero when the service policy at the multiplexer is FCFS; this symmetry disappears when a peak output rate is enforced.

In Fig. 6, the delay jitter PMF is considered for different values of $X_{\min}$, $T = 10$ and a total load of 0.9 ($\lambda^{b} = 0.8$). When $X_{\min}$ is set to $T - 1$ (in this case 9), the tagged cell stream conforms to GCRA(10, 1) (see Section II-C), and the delay jitter process possesses a very narrow region of support. On the other hand, under an FCFS policy, a significant amount of cell clustering is observed since the cell stream now "conforms" to GCRA(10, 9); cell spreading is also slightly increased.

The cost of providing a less distorted traffic stream at the multiplexer output is illustrated in Fig. 7 in terms of increased cell delays. As discussed in Section II-B, however, the effect of reducing distortion at the network premises, thereby increasing queueing delays (and buffer requirements), is essentially to distribute the buffers more uniformly throughout the network.

The remainder of this section will deal with the approximate end-to-end performance study. A network of five nodes in tandem will be considered, where the tagged traffic stream originates as a periodic source at node 1 with period $T = 5$. The background traffic at each node is assumed to consist of ten identical Bernoulli streams each with rate 0.07, resulting in a total offered load of 0.9 at each multiplexer. The relatively high load at each node, together with the small period of $T = 5$, is adopted to provide a scenario in which significant delay jitter is induced into the tagged cell stream (see Table I). The Class-1 interarrival time PMF's at nodes 2–5 will be truncated by setting ϵ equal to 10^{-6} and using the rule presented in the previous section. This choice of ϵ resulted in values of $A^{i}_{\max}$, $2 \leq i \leq 5$, ranging from 20–30.

For the results in Fig. 8, each of the five ATM multiplexers serves arriving cells according to the FCFS policy ($X_{\min} = 1$). The PMF's for interdeparture processes at nodes 1 and 5 are computed and compared with simulation results for the identical tandem queueing network. At the output of node 1, the cell interdeparture process X^{1} has experienced both clustering and spreading, but over 35% of this traffic remains periodic with $T = 5$. As the tagged stream passes through additional FCFS multiplexing stages, the interdeparture times become more and more uniformly distributed, and at the output of node 5, there are no dominant terms in the PMF. This implies that, in a real-time scenario, significant buffering may be required at the end user in order to achieve an acceptable playback quality. Also, a traffic profile such as that at the output of node 5 will likely induce more severe queueing problems (and possibly QoS degradation) for other users sharing the same network resources.

The validity of an i.i.d. assumption for cell interdepartures (and interarrivals) is addressed through the simulation results plotted in Fig. 8 as discrete points. Clearly, under an FCFS policy at each node, this assumption does not significantly affect the accuracy of the end-to-end study, as the numerical results seem to be in good agreement with simulations. Similar agreement is present in Figs. 9 and 10, where the traffic has been shaped by the peak-rate enforcing policy at each node.

In Fig. 9, each multiplexer enforces a minimum interde-

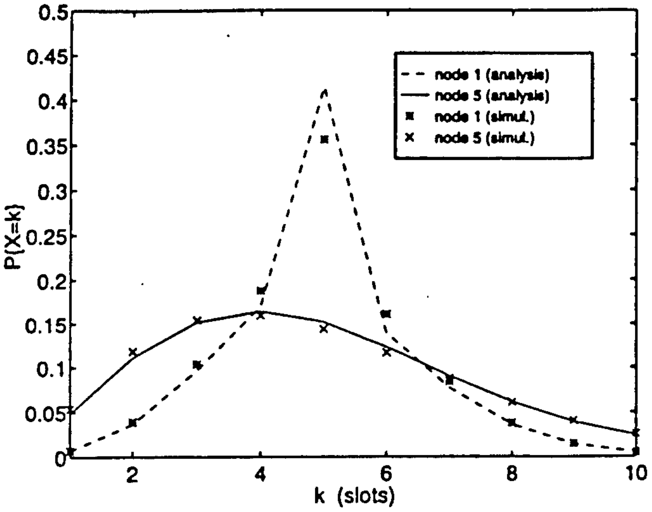


Fig. 8. Cell interdeparture time PMF's at nodes 1 and 5 under FCFS policies ($X_{\min} = 1$) at each node; $T = 5$.

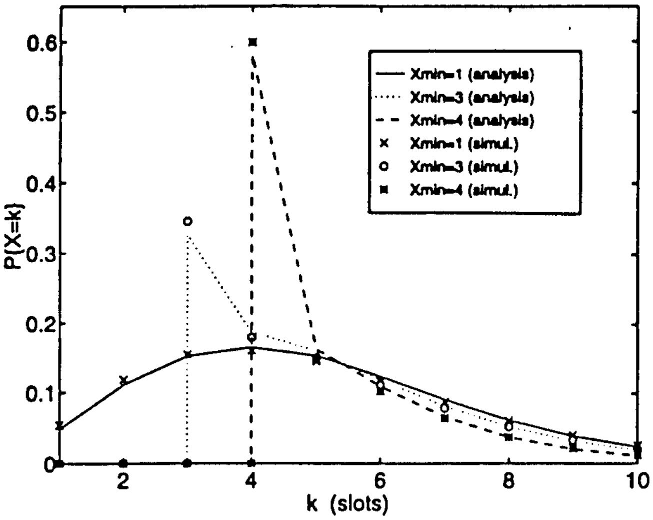


Fig. 10. Cell interdeparture time PMF's at node 5 for different $X_{\min}$.

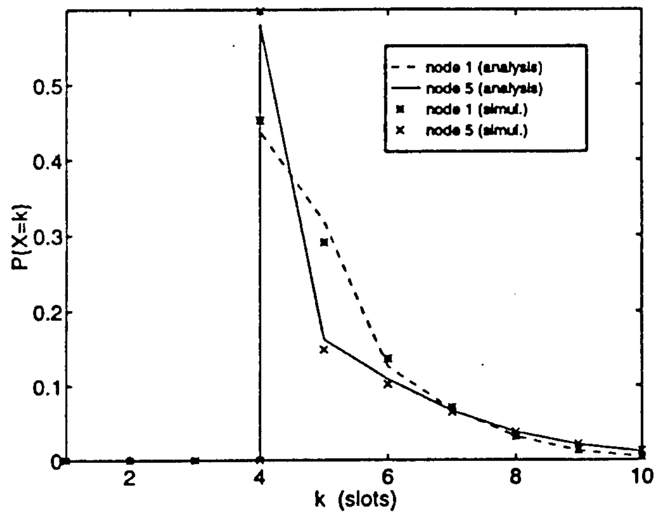


Fig. 9. Cell interdeparture time PMF's at nodes 1 and 5 for $X_{\min} = 4$ at each node; $T = 5$.

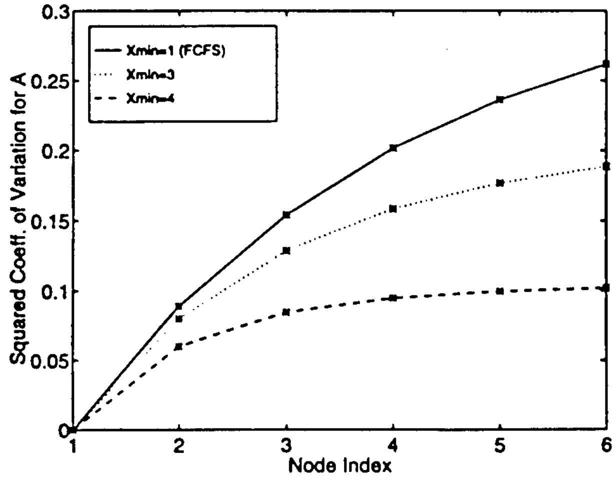


Fig. 11. Squared coefficient of variation for the cell interarrival process to each node.

parture time of $X_{\min} = 4$, and the resulting PMF's for X^1 and X^5 are presented. In both cases, the tagged stream conforms to GCRA(5, 1) throughout the network. That is, all cell clustering (or negative delay jitter) less than −1 is eliminated, and the PMF at node 5 has a much more narrow region of support than the corresponding PMF under the FCFS policy. This can be observed in Fig. 10, where three networks of servers with $X_{\min}$ equal to 1, 3, and 4 are considered. As $X_{\min}$ increases (and the CDVT parameter τ decreases), the end-to-end performance of the tagged stream, in terms of the induced distortion, improves dramatically. It is also interesting to note that the approximate analysis agrees very well with simulation results, which validates the i.i.d. assumption for cell interdepartures even when the corrective peak output-rate enforcing queueing discipline is adopted.

The squared coefficient of variation for the Class-1 interarrival process A^i is plotted in Fig. 11 as a function of the node index i for the three networks of servers considered in Fig. 10. Note that because the arrival process to node 1 is periodic, $C_A^2 = 0.0$ for node 1 in each case. As expected, C_A^2 increases with the node index for all values of $X_{\min}$. However, depending upon the peak rate being enforced by each of the servers, the rate of increase, as well as the magnitude, of C_A^2 varies greatly. Under an FCFS policy at each server, the tagged cell stream is changing significantly as it enters node 6, and the induced delay jitter is quite large. A noticeable improvement in the variability of the tagged cell stream is observed when $X_{\min}$ increases or the CDVT parameter τ decreases. In fact, when $X_{\min} = 4$ at each of the PORE servers, C_A^2 is always smaller than 0.1. The cumulative mean queueing delay for the tagged stream is plotted as a function of the node index in Fig. 12. The total mean delay appears to increase linearly with the node index, and the rate of increase is clearly dependent upon the value of $X_{\min}$. These results indicate that for real-time applications, the tradeoff between violating deadline constraints and violating either delay jitter constraints or a declared peak rate needs to be considered carefully.

Note that while delay distributions for the tagged stream are

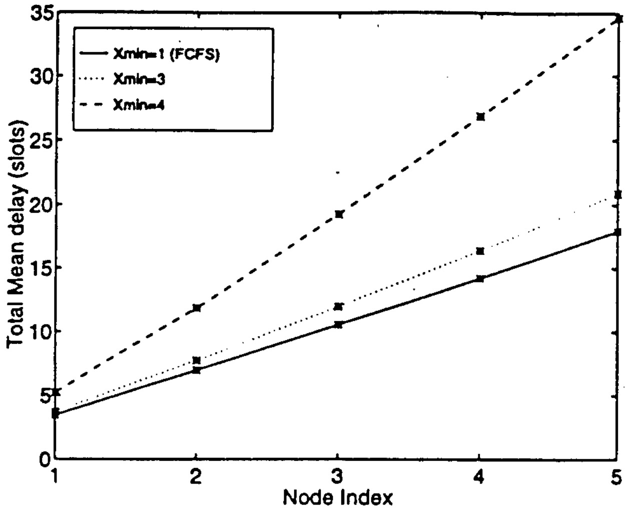


Fig. 12. Total mean queueing delay as a function of node index.

computed at each multiplexing stage, it is not yet clear how the end-to-end delay distribution can be computed numerically. By assuming that the cell delay process is independent from one node to the next, a simple convolution of individual delay PMF's would yield the approximate end-to-end delay distribution. Note that this assumption is not necessary for the computation of first moments (Fig. 12), where the mean end-to-end delay is simply the sum of individual mean delays. Although results are not presented here, the above approximation has been examined by comparing numerical and simulation results. It seems that under a network of FCFS servers, the assumption of internodal independence for the delay process yields quite accurate results. However, when peak-rate enforcement is employed at each multiplexing stage, the internodal independence assumption breaks down and the convolution of delay PMF's proves to be a poor approximation for the end-to-end delay distribution.

Finally, it should be noted that the main contribution of this work is to study the impact of reregulating on a traffic stream, analyze the PORE policy, compare this policy's impact to that under the FCFS policy, and present results for a perfectly regulated traffic (periodic traffic) to make the impact of reregulation easier to interpret. This work is directly applicable to the case in which the peak rate of one traffic stream is being enforced; this would be the case when one jitter-sensitive traffic stream is multiplexed with jitter-insensitive traffic. If a small number of jitter-sensitive traffic streams are present (or simply, peak rate is to be enforced in a small number of traffic streams), some scheduling conflicts may arise. Due to these conflicts, a greater than desirable spacing may be imposed sometimes, potentially adding to the positive jitter (more spreading), without affecting the enforced peak rate (minimum spacing). This work will still be useful in such an environment. When a large number of tagged streams are present, conflicts may dominate, and although a peak rate can still be enforced, the positive jitter and the delay may be increased significantly. In this case, the relative shifting of the periods (if the streams are periodic) is expected to have a significant impact [22].

V. Conclusions

In this paper, a queueing system was formulated to study the distortion induced in a tagged cell traffic stream; both FCFS and peak output rate enforcing (PORE) service disciplines were considered. In fact, the FCFS policy was shown to be a special case of the PORE discipline. At the output of the delay jitter-reducing service strategy, the tagged cell stream is easily characterized in accordance with the GCRA specified in [18]. A discrete-time analysis in the $M/G/1$ paradigm yielded numerical results for cell delay, delay jitter, and interdeparture time probability distributions at a multiplexer served according to the PORE policy.

An approximate end-to-end performance study of the tagged cell stream was also carried out by analyzing a tandem queueing network of PORE servers. Numerical results indicate that a peak rate enforced at the network edge can be severely violated within the network, and thus, traffic reregulation may be necessary. As intended, the traffic profile of a cell stream conforming to GCRA(T, τ) was shown to be much more regular, or less variable, when the CDVT parameter τ is small (close to zero). The tradeoff for enforcing small τ was also clearly illustrated through numerical results in the form of increased cell delays.

Simulation results for the end-to-end tandem queueing study indicate that the assumption of an i.i.d. interarrival process, with a PMF computed through the numerical approach, does not significantly affect the accuracy of the solution. The good agreement between simulation and numerical results was observed for a tandem network of FCFS servers as well as PORE servers. It is not yet clear how this numerical accuracy will be affected when a network of queues with cross-correlated traffic and different routing realizations is simulated.

VI. Appendix
Solution of the Stationary Probability Vector for P

This Appendix outlines the steps involved in computing the stationary probability vector for $\mathbf{P}$, given by $\boldsymbol{\pi} = [\boldsymbol{\pi}_0 \ \boldsymbol{\pi}_1 \ \boldsymbol{\pi}_2 \ \cdots]$. The vectors $\boldsymbol{\pi}_i$, $i \geq 0$, are the stationary probability vectors associated with level i as defined in Section II-D. The boundary probability vector $\boldsymbol{\pi}_0$ is evaluated first based on the matrix analytic techniques of Neuts [21]. The vectors $\boldsymbol{\pi}_i$, $i \geq 1$, are then recursively computed from $\boldsymbol{\pi}_0$.

A. Computation of $\boldsymbol{\pi}_0$

A brief outline of the steps involved in computing π_0 is presented here; a detailed description of the method can be found in [21]. To apply the method the following matrices need to be defined:

$$\mathbf{A} = \sum_{n=0}^{\infty} \mathbf{A}_n \tag{18}$$

$$\mathbf{G}(z) = z \sum_{n=0}^{\infty} \mathbf{A}_n \mathbf{G}^n(z), \qquad |z| \leq 1 \tag{19}$$

$$K(z) = z\sum_{n=0}^{\infty} B_n G^n(z), \qquad |z| \le 1. \qquad (20)$$

The computation of matrix $G \triangleq G(z)|_{z=1}$, which satisfies the nonlinear matrix equation

$$G = \sum_{n=0}^{\infty} A_n G^n, \qquad (21)$$

is a key step in the application of the technique. Let g denote the stationary probability vector associated with G. That is, g is the solution to $gG = g$ and $ge = 1$, where e denotes the unit column vector.

Matrix G can be obtained as the limit of the monotonically increasing, convergent matrix sequence $\{G_k\}_{k\ge 0}$, defined by

$$\begin{aligned} G_0 &= 0, \\ G_{k+1} &= \sum_{n=0}^{\infty} A_n G_k^n \qquad k \ge 0. \end{aligned} \qquad (22)$$

Matrix G is then used to determine matrix K, defined by $K = K(1)$, and given by

$$K = \sum_{n=0}^{\infty} B_n G^n \qquad (23)$$

with stationary probability vector κ.

Let κ^* be the vector defined as

$$\kappa^* = \left.\frac{dK(z)}{dz}\right|_{z=1} e.$$

The expression for κ^* is obtained by differentiating (20) at $z = 1$:

$$\begin{aligned} \kappa^* &= \left.\frac{dK(z)}{dz}\right|_{z=1} e \\ &= \sum_{n=0}^{\infty} B_n G^n e + \sum_{n=1}^{\infty} B_n \\ &\quad \cdot \sum_{k=0}^{n-1} G^k \left.\frac{dG(z)}{dz}\right|_{z=1} G^{n-1-k} e \\ &= Ke + \sum_{n=1}^{\infty} B_n \sum_{k=0}^{n-1} G^k \mu \end{aligned} \qquad (24)$$

where $\mu = [dG(z)/dz]|_{z=1} e$ and is given by

$$\mu = (I - G + eg)[I - A + (e - \beta)g]^{-1} e. \qquad (25)$$

The vector β is given by

$$\beta = \sum_{n=1}^{\infty} n A_n e. \qquad (26)$$

Finally, the stationary probability vector π_0 can be expressed in terms of the vectors κ and κ^* as

$$\pi_0 = \frac{\kappa}{\kappa\kappa^*}.$$

B. Computation of π_i, $i \ge 1$

The vectors π_i, for $i \ge 1$, can now be computed using the recursive scheme attributed to Ramaswami [23],

$$\begin{aligned} \pi_i = &\left[\pi_0 B_i^* + \sum_{j=1}^{i-1} \pi_j A_{i+1-j}^*\right] \\ &\cdot (I - A_1^*)^{-1}, \qquad i \ge 1 \end{aligned} \qquad (27)$$

where

$$\begin{aligned} B_n^* &= \sum_{k=n}^{\infty} B_k G^{k-n}, \\ A_n^* &= \sum_{k=n}^{\infty} A_k G^{k-n}, \qquad n \ge 0. \end{aligned}$$

REFERENCES

[1] I. Stavrakakis, "Efficient modeling of merging and splitting processes in large networking structures," *IEEE J. Select. Areas Commun.*, vol. 9, no. 8, Oct. 1991.

[2] W. Matragi, C. Bisdikian, and K. Sohraby, "Jitter calculus in atm networks: Part I, Single node," in *Proc. IEEE INFOCOM'94*, Toronto, Canada, 1994.

[3] J. Roberts and F. Guillemin, "Jitter in ATM networks and its impact on peak rate enforcement," *Perf. Eval.*, vol. 16, nos. 1–3, Nov. 1992.

[4] R. Cruz, "A calculus for network delay, Part I: Network elements in isolation," *IEEE Trans. Inform. Theory*, vol. 37, no. 1, Jan. 1991.

[5] ——, "A calculus for network delay, Part II: Network analysis," *IEEE Trans. Inform. Theory*, vol. 37, no. 1, Jan. 1991.

[6] A. Parekh and R. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single-node case," *IEEE/ACM Trans. Networking*, vol. 1, no. 3, June 1993.

[7] ——, "A generalized processor sharing approach to flow control in integrated services networks: The multiple node case," *IEEE/ACM Trans. Networking*, vol. 1, no. 3, June 1993.

[8] D. Towsley, "Providing quality of service in packet switched networks," in *Perf. Eval. Comput. Commun. Syst.* (Joint Tutorial Papers of PERFORMANCE'93 and SIGMETRICS'93), 1993.

[9] J. Kurose, "On computing per-session performance bounds in high-speed multi-hop computer networks," in *Proc. ACM SIGMETRICS'92*.

[10] O. Yaron and M. Sidi, "Calculating performance bounds in communication networks," in *Proc. IEEE INFOCOM'93*, San Francisco, CA.

[11] R. Guerin *et al.*, "Equivalent capacity and its application to bandwidth allocation in high-speed networks," *IEEE J. Select. Areas Commun.*, vol. 9, no. 7, Sept. 1991.

[12] D. Ferrari and D. Verma, "A scheme for real-time channel establishment in wide-area networks," *IEEE J. Select. Areas Commun.*, vol. 8, no. 3, Apr. 1990.

[13] J. Hyman *et al.*, "Real-time scheduling with quality of service constraints," *IEEE J. Select. Areas Commun.*, vol. 9, no. 7, Sept. 1991.

[14] W. Matragi, K. Sohraby, and C. Bisdikian, "Jitter calculus in atm networks: Part ii, multiple nodes," in *Proc. IEEE INFOCOM'94*, Toronto, Canada.

[15] H. Kroner, M. Eberspacher, T. Theimer, P. Khun, and U. Briem, "Approximate analysis of end-to-end delay in ATM networks," in *Proc. IEEE INFOCOM'92*, Florence, Italy.

[16] J. F. Ren, J. Mark, and J. Wong, "End-to-end performance in atm networks," in *Proc. IEEE ICC'94*, New Orleans, LA.

[17] Y. Ohba, M. Murata, and H. Miyahara, "Analysis of interdeparture processes for bursty traffic in ATM networks," *IEEE J. Select. Areas Commun.*, vol. 9, no. 3, Apr. 1991.

[18] ATM Forum, *ATM User-Network Interface Specification*, Ver. 3.0, 1993.

[19] C. Bisdikian, W. Matragi, and K. Sohraby, "A study of the jitter in atm multiplexers," in *Proc. 5th Int. Conf. Data Commun. Syst. Perf.*, Raleigh, NC, 1993.

[20] F. Guillemin and W. Monin, "Management of cell delay variation in ATM networks," in *Proc. IEEE GLOBECOM'92*, Orlando, FL.

[21] M. F. Neuts, *Structured Stochastic Matrices of the M/G/1 Type and Their Applications*. New York: Marcel Dekker, 1989.

[22] R. Landry and I. Stavrakakis, "Non-deterministic periodic packet streams and their impact on a finite-capacity multiplexer," in *Proc. IEEE INFOCOM'94*, Toronto, Canada.

[23] V. Ramaswami, "A stable recursion for the steady state vector in Markov chains of the M/G/1 type," *Commun. Statist.—Stochastic Models*, vol. 4, no. 1, 1988.

Randall Landry (M'94) received the M.S. and Ph.D. degrees in electrical engineering from the University of Vermont. He is a research scientist in the Network Technology Branch of the DSP R&D Center at Texas Instruments, Dallas. Dr. Landry is currently engaged in research efforts to develop very high-speed, highly integrated switch architectures for Gigabit networking. His research interests also include multi-layer switching technologies, Quality of Service provisioning in the Internet and the internetworking of connectionless and connection-oriented protocols such as Ethernet and ATM.

Ioannis Stavrakakis (SM'93) received the Diploma in electrical engineering from the Aristotelian University of Thessaloniki, Thessaloniki, Greece, 1983, and the Ph.D. degree in electrical engineering from the University of Virginia, 1988. In 1988, he joined the faculty of Computer Science and Electrical Engineering at the University of Vermont as an assistant and then associate professor. Since 1994, he has been an associate professor of Electrical and Computer Engineering at Northeastern University, Boston. His research interests are in stochastic system modeling, teletraffic analysis, and discrete-time queueing theory, with primary focus on the design and performance evaluation of Broadband Integrated Services Digital Networks (B-ISDN). Dr. Stavrakakis is a Senior Member of IEEE and a member of the IEEE Communications Society, Technical Committee on Computer Communications. He has organized and chaired sessions, and has been a technical committee member, for conferences such as GLOBECOM, ICC, and INFOCOM.

Evaluation of ABR Traffic Management Under Various System Time Scales [*]

Tamer Dağ   Ioannis Stavrakakis
Electrical and Computer Engineering Department
409 Dana Research Building, 360 Huntington Avenue
Northeastern University, Boston, MA 02115
e-mail: ioannis@hilbert.cdsp.neu.edu

Abstract

In this paper, the effects of various time scales on the management of ABR (Available Bit Rate) traffic using feedback based control is studied. Since delay tolerable, the ABR applications can be allocated the remaining resources after CBR (Constant Bit Rate) and VBR (Variable Bit Rate) applications have been accommodated. To avoid excessive losses the transmission rate of the ABR applications should be modulated by the amount of remaining resources. That is, the ABR rate should be controlled through a feedback based rate control mechanism. In this paper a network link shared by remote ABR and VBR applications is considered and the impact of various system time scales on the effectiveness of the feedback based rate control scheme is investigated, by formulating and studying a simple analytical model. These time scales are expressed in terms of the network transmission speed, the minimum tolerable ABR rate and the rate of change of the VBR rate. While the negative impact of a decreased network time scale on the effectiveness of this control scheme is well established, the impact of the ABR and VBR time scales has not been investigated in the past. It turns out that for a given network time scale, the induced cell losses can be significantly reduced for increased ABR and/or VBR time scales and thus, the latter time scales should be taken into consideration when evaluating the effectiveness of an adaptive feedback based congestion control mechanism. This study also suggests that higher efficiency can be achieved by enforcing large ABR time scales, leading to the introduction of a new class of transmission policies.

[*] This research is supported in part by the Advanced Research Project Agency under Grant F49620-93-1-0564 monitored by the Air Force Office of Scientific Research (AFOSR) and the National Science Foundation under Grant NCR-9628116.

1 Introduction

Asynchronous Transfer Mode (ATM) has been chosen as the CCITT standard for the switching and multiplexing technique of the Broadband-Integrated Services Digital Networks (B-ISDN) currently under development. ATM networks are expected to support applications with diverse traffic characteristics and Quality of Service (QoS) requirements, such as data transfer, video, voice, multi-media conference and real time control.

ATM is primarily a connection oriented protocol. Virtual circuits are set and the route is decided at connection set up. Connection oriented services have typically long sustained sessions and cells are transmitted and delivered in order. Therefore, it appears like a dedicated link where all cells associated with a service follow the same path. Statistical multiplexing is possible by on-demand transmitting cells from different sources over the same line and storing them temporarily in shared buffers within the network.

The ATM Forum [1] has defined four different service classes. Constant Bit Rate (CBR), Variable Bit Rate (VBR), Available Bit Rate (ABR) and Unspecified Bit Rate (UBR). These classes can cover a broad spectrum of diverse applications. Non-real time VBR, ABR and UBR service classes are intended for non-real time applications, while CBR and real time VBR are intended for real time applications.

The ABR service class is intended for non-real time applications which can tolerate delay. This service class simply uses the remaining capacity left from VBR and CBR services and is a best effort service. The goal of this service is to use the unused network bandwidth as efficiently as possible. For ABR the only negotiated QoS parameter is the Cell Loss Ratio (CLR) and is not necessarily quantified. Peak Cell Rate (PCR), CDVT and Minimum Cell Rate (MCR) are the traffic parameters.

The ABR service class is different from the other service classes in many ways [2] . The available bandwidth will be shared by the ABR connections and will change dynamically due to the fluctuations in the VBR and CBR traffic and it may reduce to the MCR. Since VBR and CBR traffic rates will fluctuate in time, the ABR sources should adjust their rates to these fluctuations in order to utilize the entire remaining capacity or avoid cell losses.

These characteristics point to the necessity of controlling the flow of the ABR traffic sources. Two major flow control mechanisms for the ABR service have been considered: the rate based control [3], [4], [5] and the credit based control [6], [7], [8]. Since both of these control schemes have advantages and disadvantages, there were also studies on the integration [9] of both schemes.

The credit based flow control is implemented link by link and per virtual circuit (VC). Before transmitting any cells over the link, the sender associated with the VC link needs to receive credits for the VC from the receiver. The credit indicates the amount of free buffer space available for the sender. The sender can not transmit more cells than its credit. If it uses all its credits, it must wait for other credits. Because of this, credit based flow control ensures zero cell loss.

In rate based flow control, the sender is notified about congestion in the network with a feedback generated from the receiver or from the congested nodes. When the sender receives such a feedback, it adjusts its rate according to the feedback control policy in effect. Although the credit based flow control can handle both smooth and bursty traffic (while the rate based flow control may not be very effective for bursty traffic), it introduces an excessive amount of overhead and it would be very expensive for a wide area network. For such reasons, the more flexible rate based flow control has been adopted by the ATM Forum.

Many different traffic control policies have been proposed for the rate based flow control such as the Explicit Forward Congestion Indication (EFCI) [10], Backward Explicit Congestion Notification (BECN) [11], Proportional Rate Control Algorithm (PRCA) [12], Explicit Rate Feedback [13] and Enhanced Proportional Rate Control Algorithm (EPRCA) [14].

If the propagation delay is non-zero, the ABR sources will learn about the change in the rate of the VBR/CBR sources - that is, the current level of the available bandwidth - only after they have received the feedback information. Meanwhile, the ABR sources will transmit at a rate which is based on the most recent level of available bandwidth known to them. Consequently, the effectiveness of the rate based flow control schemes will decrease under non-zero propagation delays.

For example, if the propagation delay between an ABR source and the network access node is T_d and the available bandwidth changes at time t, the ABR source will learn about the change in bandwidth availability at time $t+T_d$. The adjusted rate from the ABR source will reach the network access node at time $t+2T_d$, assuming that the backward and forward propagation delays are equal. This can be clearly observed in Figure 1 where a network link is shared by CBR, VBR and ABR applications. Whenever the VBR/CBR sources change their rates, a bandwidth mismatch occurs for a duration of $2T_d$.

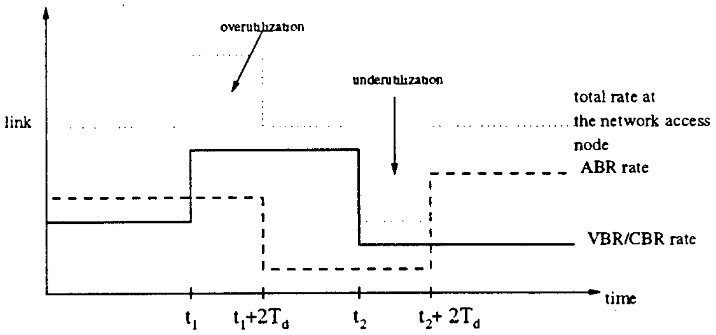


Figure 1: An example of bandwidth mismatch

When a decrease in the remaining bandwidth occurs, the link capacity is exceeded for a duration of $2T_d$ and overutilization occurs. During an overutilization period, the number of cells in transit becomes excessive and this causes cell losses. When an increase in the remaining bandwidth occurs, underutilization occurs for a duration of $2T_d$. While the effect of increased propagation delay on the flow control schemes has been studied in the past [15], [16], [17], almost no effort [18] has been focused on the study of the impact of other system parameters. In this paper, the impact of these system parameters (network, ABR and VBR time scales) on the feedback based flow control is investigated.

The network time scale is defined as the transmission time of a cell and it decreases as the network speed increases. Therefore, decreased network time scale corresponds to increased propagation delay. The VBR time scale can be measured from the rate of change of the VBR source transmission rate and the ABR time scale is defined as the minimum distance among subsequent blocks of subsequent cells generated by the ABR source. These time scales are defined in detail later.

The remainder of the paper is organized as follows. In section 2, the system considered in order to evaluate the impact of time scales is described, In section 3, a suitable Markov Chain for the system is formulated and some numerical results are presented in section 4. The paper ends with the conclusions made in section 5.

2 Description of the System

Consider a transmission link shared by Variable Bit Rate (VBR) and Available Bit Rate (ABR) applications. The VBR sources are provided prioritized access to the resources and ABR sources are allocated the remaining resources. In order to simplify the analysis and facilitate the understanding of the impact of the various time scales, a system with one VBR source and one ABR source is considered. Each of these sources will deliver at most one cell per slot according to the rate in effect.

The ABR traffic source is assumed to be away from the network access node and thus the propagation delay between the ABR source and the network access node is nonnegligible. The distance between the VBR source and the network access node is not important since no control is applied on the VBR source. The ABR source is controllable. Its transmission rates are calculated by the network access node by taking into consideration the current VBR transmission rates. The ABR source rates should be calculated appropriately in order not to cause instability. Therefore, the total ABR and VBR source rates at any time instant should not exceed the link capacity.

The network access node is supposed to be able to detect a change in the VBR source rates. This may be possible through an explicit indication that the VBR source carries by certain cells or an estimation mechanism implemented at the network access node. When a rate change for the VBR sources occur, the network access node detects it and calculates the appropriate transmission rates for the ABR sources. Then, this information is fed back to the ABR sources.

Because of the nonzero distance d between the ABR traffic sources and the network access node, bandwidth mismatch will occur when VBR traffic rate changes occur. A change in the VBR rate at time t will be detected by the ABR source at time $t + T_d$ and the adjusted ABR rate will reach the network access node at time $t + 2T_d$. During the above round-trip propagation delay $2T_d$, the network may either be underutilized or overutilized; overutilization is the main reason causing cell losses.

The network access node is assumed to have a finite buffer of capacity C for the temporary storage of the ABR cells; a buffer of capacity 1 is sufficient for the temporary storage of the VBR cells, since the one service per slot will be available to the VBR cell upon its arrival to the network access node.

A queuing model of the network access node is shown in Figure 2. The service policy is assumed to be work conserving and VBR cell arrivals are provided head of line (HoL) priority as indicated above. ABR cells are served according to the first come first served (FCFS) policy. Cell departures are assumed to occur before cell arrivals occurring over the same slot.

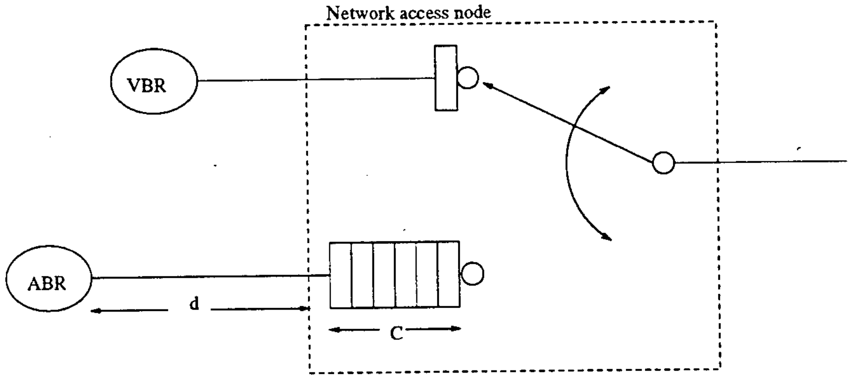


Figure 2: The queuing model of the network access node

2.1 The ABR Traffic Source

It is assumed that the ABR source can transmit at the maximum allowable rate selected for it by the network access node. Thus, a heavy traffic assumption is made. Under this assumption the ABR source will always have a cell to transmit with its allowed rate. Therefore, the worst case performance of the ABR source can be derived. The maximum allowable ABR source rate can be calculated from the current VBR source rate by the network access node. For example, if the current VBR source rate is equal to r_v, then the maximum allowable ABR source rate will be equal to $1 - r_v - \epsilon$ resulting in a total link rate equal to $1 - \epsilon$. Here, ϵ is an arbitrarily small positive number to ensure that the maximum system load is below 1 (stability condition).

In order to satisfy this rate, the ABR source can transmit one cell per T slots where $T = \lceil \frac{1}{1-r_v-\epsilon} \rceil$ and $\lceil () \rceil$ denotes the smallest integer larger than the (). T is called the *fundamental ABR subframe*. The same rate can also be achieved, if the ABR source transmits n cells per subframe of length nT slots, that is, if the source employs an *n-order subframe*.

Let t_k denote the beginning of the k^{th} subframe of length B_kT_k. T_k is equal to the fundamental subframe which is equal to $\lceil \frac{1}{1-r_{v_k}-\epsilon} \rceil$ and r_{v_k} is the effective VBR source rate at t_k; B_k ($B_k \in N$), denotes the length of the batch transmitted over the k^{th} subframe. If $B_k = 1$, the ABR source will employ the fundamental subframe T_k to transmit at a rate equal to $1/T_k$. If $B_k > 1$, the ABR source will employ a B_k-order subframe to transmit at the same rate equal to $1/T_k$. Note that, the ABR source time scale is equal to the length of the *B_k-order subframe* (B_kT_k). Figure 3 shows examples of transmission employing fundamental or higher-order subframes.

Let $TC(n, nT)$ denote a transmission control policy according to which the ABR source transmits a batch of size n at the beginning of a subframe of length nT. Let $\overline{TC}(T) = \{TC(n, nT), n \geq 1\}$ denote the class of all $TC(n, nT)$ policies which implement the same transmission rate of $1/T$. This class of policies is characterized by the fundamental subframe of length T.

At a subframe boundary t_k a fundamental subframe T_k (and ABR rate) will be determined based on the current VBR rate r_{v_k}. Any policy in $\overline{TC}(T_k)$ can be selected for the implementation of the ABR rate of $1/T_k$. One of the objectives of this paper is to study the performance of the various policies within a given class (the different ABR time scales) and provide guidelines regarding the selection of the optimal one.

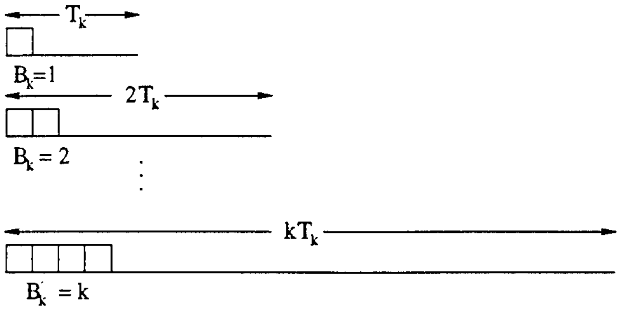


Figure 3: The ABR traffic subframes for a transmission rate of $1/T_k$

2.2 The VBR Traffic Source

Although the activity level of the VBR traffic source is not controllable and can change in principle at any slot, it will be assumed that such changes occur only at subframe boundaries for analysis tractability. Therefore, for the VBR source, time is considered to be subframed and VBR source rate changes can occur anywhere in this subframed time.

Let $\{S_k\}_{k\geq 1}$ be a 2-state underlying Markov chain with state space $\overline{S} = \{0,1\}$. $\{S_k\}_{k\geq 1}$ will be used to describe the VBR source activity level. $\{S_k\}_{k\geq 1}$ is embedded at subframe boundaries $\{t_k\}_{k\geq 1}$ of length $B_k T_k$, where T_k is determined by the perceived remaining capacity as shown in the previous subsection.

Let $r_v(s_k)$ denote the rate of the VBR source at the k^{th} subframe, $s_k \in S$. Without loss of generality, it is assumed that $r_v(1) > r_v(0)$. The VBR cell arrival process is modeled as a Markov Modulated Bernoulli process. The probability of transmitting 1(0) cell in a slot of the k^{th} subframe is $r_v(s_k)(1 - r_v(s_k))$.

Let $p_s(s_k)$ denote the probability that S_k changes at the end of the k^{th} subframe at the beginning of which it was in state s_k. $p_s(s_k)$ will reflect the average time over which VBR applications maintain a constant rate (or VBR time scale). If $p_s(0) = p_s(1) = p_s$, then the VBR time scale will be equal to $1/p_s$; otherwise, two time scales may be defined.

3 Analysis of the System

In this section a proper Markov Chain describing the evolution of the system is described by taking into consideration the properties of the ABR and VBR sources described in the previous section. Then, a numerical approach is developed for the evaluation of the loss performance of the system.

Since, no VBR cell losses will occur, the analysis is focused on the evaluation of the loss performance of the ABR source.

Let $\{S_k, Q_k\}_{k\geq 1}$ be a 2-dimensional process embedded on $\{t_k\}_{k\geq 1}$, where Q_k is a random variable describing the buffer occupancy at t_k, $0 \leq Q_k \leq C$. By definition, S_k determines completely the interfering or the VBR arrival process.

If the propagation delay T_d is negligible, then S_k determines the ABR cell arrival process as well, because the ABR source will respond to the VBR source rate changes immediately. If the propagation delay is nonnegligible - as it is considered to be the case in this work - the *fundamental ABR subframe* length can be calculated from $T_k = \lceil \frac{1}{1-r_v(s'_k)-\epsilon} \rceil$ where s'_k is the VBR state $2T_d$ time units earlier. As a consequence, the current state S_k does not determine the arrival process and thus $\{S_k, Q_k\}_{k\geq 1}$ is not a Markov chain.

Although a Markov chain can be constructed to describe the evolution of the system under fixed and non-zero roundtrip propagation delays, a simplifying assumption is made by assuming that the roundtrip propagation delay is random and geometrically distributed with mean $\frac{1}{p_f}$ measured in terms of subframes. Given that a VBR source rate change has occured, a feedback indicating the VBR source rate change will arrive at the ABR source by the next subframe boundary with probability p_f. Note that, p_f is adjusted based on the current subframe length so that the mean propagation delay under both subframe schemes are equal.

In order to determine the arrival process to the node over the k^{th} subframe, the network access node needs to know the current VBR source rate ($r_v(s_k)$) and whether the current ABR rate is based on the most recent feedback sent by the node or not (that is, whether the feedback's impact is not pending or pending, respectively). Therefore, another random variable is introduced describing the status of the impact of the feedback at t_k.

Let J_k be an indicator function which assumes the value of 1, if the impact of a feedback carrying a VBR source rate change is pending at t_k; that is, if the ABR rate arriving at the node is not adjusted yet. If the impact of the feedback is not pending, J_k is equal to 0. The ABR traffic rate $\frac{1}{T_k}$ will be completely determined by (S_k, J_k) and it is given by $\frac{1}{T_k} = 1 - r_v(s_k \oplus j_k) - \epsilon$ where $\oplus$ denotes the modulo 2 addition; $i \oplus j$ will be $0(1)$ if $i = j(i \neq j)$.

Although it is possible that at t_k the impact of more than one feedbacks is pending, at most one pending feedback will be considered in order to simplify the analysis. This approximation holds true if $max_{s_k}\{p_s(s_k)\} \ll p_f$; that is, if the VBR source time scale is much larger than the propagation delay.

Under the above assumptions, it is easy to establish that the stochastic process $\{S_k, J_k, Q_k\}$ embedded at subframe boundaries is a Markov chain with state space $\{(s_k, j_k, q_k) : 0 \leq s_k \leq 1, 0 \leq j_k \leq 1, 0 \leq q_k \leq C\}$.

Let $P(s_k j_k q_k; s_{k+1} j_{k+1} q_{k+1})$ be the probability that the Markov chain $\{S_k, J_k, Q_k\}$ moves from state (s_k, j_k, q_k) at time t_k, to state $(s_{k+1}, j_{k+1}, q_{k+1})$ at time t_{k+1}. From the description of the system model, it can be understood that (s_{k+1}, j_{k+1}) is independent from the queue occupancy q_k and thus $P(s_k j_k q_k; s_{k+1} j_{k+1} q_{k+1})$ can be expressed as follows.

$$P(s_k j_k q_k; s_{k+1} j_{k+1} q_{k+1}) = P(s_k j_k; s_{k+1} j_{k+1}) P(q_{k+1}/s_k j_k q_k)$$

The probabilities on the right side of the above equation can be calculated separately as shown below. Note that $P(s_kj_k;s_{k+1}j_{k+1})$ is the probability of passing from (s_k, j_k) to (s_{k+1}, j_{k+1}) and $P(q_{k+1}/s_kj_kq_k)$ is the probability that the queue occupancy at the beginning of the next subframe is q_{k+1}, given the current subframe state $(s_kj_kq_k)$.

The state diagram for the $\{S_k, J_k\}$ process is shown in Figure 4. Notice that not all state transitions are possible because of the constraint on j_k and the requirement of generating a feedback only when the VBR rate changes. Let $(s_kj_k) \rightarrow (s_{k+1}j_{k+1})$ denote a state transition from (s_kj_k) to $(s_{k+1}j_{k+1})$. For example, a (00) → (10) transition is impossible, because a pending feedback has to be generated when a VBR rate change occurs. This transition may be possible only if the feedback delay is zero. Similarly, a transition from (11) → (01) can occur with a very low probability and it is assumed to be equal to 0. This result comes from the assumption that the VBR source time scale is much larger than the propagation delay.

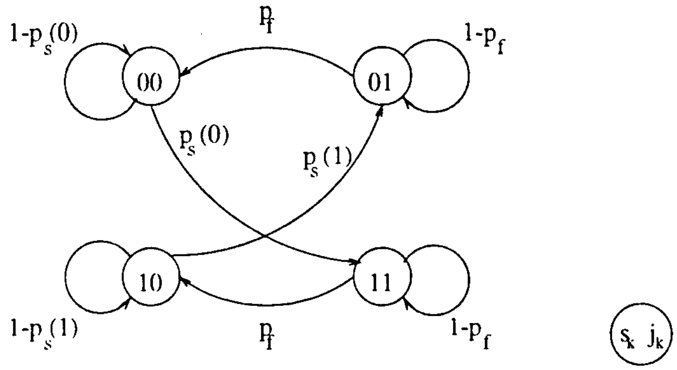


Figure 4: The state diagram of the $\{S_k, J_k\}$ process

(s_kj_k)	$(s_{k+1}j_{k+1})$	$P(s_kj_k, s_{k+1}j_{k+1})$
00	00	$1 - p_s(0)$
00	01	0
00	10	0
00	11	$p_s(0)$
01	00	p_f
01	01	$1 - p_f$
01	10	0
01	11	0
10	00	0
10	01	$p_s(1)$
10	10	$1 - p_s(1)$
10	11	0
11	00	0
11	01	0
11	10	p_f
11	11	$1 - p_f$

Let $q_{k,i}$ denote the queue occupancy at the end of the i^{th} slot of the k^{th} subframe and $P_A(q_{k,i+1}/s_kj_kq_{k,i})$ and $P_B(q_{k,i+1}/s_kj_kq_{k,i})$ denote the one slot transition probabilities from $q_{k,i}$ to $q_{k,i+1}$ over the first B_k slots of the k^{th} subframe (where both ABR and VBR cell arrivals are possible) and the remaining $B_kT_k - B_k$ slots (where only VBR cell arrivals are possible); respectively. These probabilities are equal to,

$$\mathrm{P_A(q_{k,i+1}/s_kj_kq_{k.i})} = \begin{cases} P\{A(s_k) = q_{k,i+1} - 1\} & q_{k,i} = 0, 0 \le q_{k,i+1} \le C - 1 \\ P\{A(s_k) = q_{k,i+1} - q_{k,i}\} & 1 \le q_{k,i} \le C, 0 \le q_{k,i+1} \le C - 1 \\ \sum_{m=0}^{1-C+q_{k,i}} P\{A(s_k) = C - q_{k,i} + m\} & 0 \le q_{k,i} \le C, 0 \le q_{k,i+1} \le C. \end{cases}$$

where $A(s_k)$ denotes the number of cell arrivals from the VBR traffic source in a slot when the current VBR state is s_k. Similarly,

$$\mathrm{P_B(q_{k,i+1}/s_kj_kq_{k,i})} = \begin{cases} P\{A(s_k) = q_{k,i+1}\} & q_{k,i} = 0, 0 \le q_{k,i+1} \le C \\ P\{A(s_k) = q_{k,i+1} - q_{k,i} + 1\} & 1 \le q_{k,i} \le C, 0 \le q_{k,i+1} \le C. \end{cases}$$

Combining these two equations $P(q_{k+1}/s_kj_kq_k)$ can be expressed as follows,

$$P(q_{k+1}/s_kj_kq_k) = \sum_{q_{k,B_k}=0}^{C} P_A(q_{k,B_k}/s_kj_kq_k) P_B(q_{k+1}/s_kj_kq_{k,B_k}).$$

The B_k slot transition probability $P_A(q_{k.B_k}/s_kj_kq_k)$ and $B_kT_k - B_k$ slot transition probability $P_B(q_{k+1}/s_kj_kq_{k.B_k})$ can be calculated iteratively using the 1 slot transition probabilities.

After the calculation of the transition probabilities $P(s_kj_kq_k; s_{k+1}j_{k+1}q_{k+1})$, the steady state probability distribution $\Pi(s_k, j_k, q_k)$ can be derived from the following matrix equations,

$$\begin{aligned} \Pi &= \Pi P \\ \sum_i \Pi_i &= 1. \end{aligned}$$

where P is the state transition matrix and each of its elements show the transition probabilities $P(s_kj_kq_k; s_{k+1}j_{k+1}q_{k+1})$.

In order to study the impact of network and source time scales, the ABR cell loss probability is evaluated. The ABR cell loss probability can be calculated using the average number of cells arrived at a subframe in which $\{S_k, J_k, Q_k\}_{k\ge1}$ is in state (s_k, j_k, q_k) and taking the expectation over all possible states. Therefore, the ABR cell loss probability denoted by LP is given as,

$$LP = \mathrm{E}\{ \frac{Average \ number \ of \ cells \ lost \ in \ state \ (s_kj_kq_k)}{Average \ number \ of \ cells \ arrived \ in \ state \ (s_kj_kq_k)} \}$$

Let $\overline{L}(s_k, j_k, q_k)$ and $\overline{M}(s_k, j_k, q_k)$ denote the average number of cells lost and arrived over the subframe at the beginning of which the process $\{S_k, J_k, Q_k\}_{k\ge1}$ is in state (s_k, j_k, q_k), respectively. Then,

$$LP = E\{\frac{\overline{L}(s_k, j_k, q_k)}{\overline{M}(s_k, j_k, q_k)}\}.$$

In order to calculate $\overline{L}(s_k, j_k, q_k)$ and $\overline{M}(s_k, j_k, q_k)$, $I(s_k, j_k, q_k; i)$ and $R(s_k, j_k; i)$ can be introduced. Let $I(s_k, j_k, q_k; i)(R(s_k, j_k, q_k; i))$ denote the number of cells lost(arrived) at the i^{th} slot of the subframe at the beginning of which the system is in state $(s_kj_kq_k)$; $1 \leq i \leq B_kT_k$. $R(s_k, j_k, q_k; i)$ does not depend on the queue occupancy and is a function of s_k, j_k and i since $T_k = f(s_k, j_k)$. Therefore,

$$R(s_k, j_k; i) = \begin{cases} A(s_k) & \text{if } i > B_k \\ A(s_k) + 1 & \text{if } i \leq B_k. \end{cases}$$

The number of lost cells can be found by considering the number of arrivals, the buffer occupancy in the previous slot and the buffer capacity C. Therefore,

$$I(s_k, j_k, q_k; i) = \begin{cases} m & \text{if } R(s_k, j_k; i) = C + 1 - q_{k,i-1} + m \\ 0 & \text{otherwise }. \end{cases}$$

Clearly, the number of cell arrivals over a subframe at the beginning of which the system is in state $(s_kj_kq_k)$ is equal to the summation of $R(s_k, j_k; i)$ for all slots i of this subframe. Thus,

$$\begin{aligned} M(s_k, j_k, q_k) &= \sum_{i=1}^{B_kT_k} R(s_k, j_k, q_k) \\ &= \sum_{i=1}^{B_k}(A(s_k) + 1) + \sum_{i=B_k+1}^{B_kT_k} A(s_k) \\ &= B_k + B_kT_kA(s_k). \end{aligned}$$

Taking the expectation of both sides,

$$\overline{M}(s_k, j_k, q_k) = E\{B_k + B_kT_kA(s_k)\} = B_k + B_kT_kr_v(s_k).$$

The total number of cells lost over the entire subframe is equal to the summation of $I(s_k, j_k, q_k; i)$ for all slots i of this subframe.

$$L(s_k, j_k, q_k) = \sum_{i=1}^{B_kT_k} I(s_k, j_k, q_k; i).$$

Taking the expectation of both sides,

$$\overline{L}(s_k, j_k, q_k) = \sum_{i=1}^{B_kT_k} E\{I(s_k, j_k, q_k; i)\}.$$

The above equation can be divided into two terms, the first term corresponding to the first part of the subframe (when both ABR and VBR cell arrivals are possible) and the second term corresponding to the second part of the subframe (when only VBR cell arrivals are possible). Thus,

$$\begin{aligned} \overline{L}(s_k, j_k, q_k) &= \sum_{i=1}^{B_k} E\{I(s_k, j_k, q_k; i)\} \\ &+ \sum_{i=B_k+1}^{B_kT_k} E\{I(s_k, j_k, q_k; i)\}. \end{aligned}$$

By employing the definitions for $I(s_k, j_k, q_k; i)$ and $R(s_k, j_k; i)$ the following expression is obtained,

$$\begin{aligned}\overline{L}(s_k, j_k, q_k) &= \sum_{i=1}^{B_k}\sum_{m} mP\{A(s_k) = C - q_{k,i-1} + m/q_k, s_k, j_k\} \\ &+ \sum_{i=B_k+1}^{B_kT_k}\sum_{m} mP\{A(s_k) = C + 1 - q_{k,i-1} + m/q_k, s_k, j_k\}.\end{aligned}$$

Since there are at most one arrival per slot after the first B_k slots of each subframe, cell loss may occur only in the first B_k slots. Therefore,

$$\overline{L}(s_k, j_k, q_k) = \sum_{i=1}^{B_k}\sum_{m} mP\{A(s_k) = C - q_{k,i-1} + m/q_k, s_k, j_k\}$$

and

$$\overline{L}(s_k, j_k, q_k) = \sum_{i=1}^{B_k}\sum_{l=0}^{C}\sum_{m=0}^{l-C+1} mP\{A(s_k) = C - l + m/q_k, s_k, j_k\}P\{q_{k,i-1} = l/q_{k,0} = q_k\}.$$

Combining $\overline{L}(s_k, j_k, q_k)$ and $\overline{M}(s_k, j_k, q_k)$ equations obtained above, the ABR cell loss probability, LP, can be expressed as follows,

$$\begin{aligned}LP &= \sum_{s_k=0}^{1}\sum_{j_k=0}^{1}\sum_{q_k=0}^{C}\frac{\Pi(s_k, j_k, q_k)}{B_k + r_v(s_k)B_kT_k} \\ &\times \sum_{i=1}^{B_k}\sum_{l=0}^{C}\sum_{m=0}^{l-C+1} mP\{A(s_k) = C - l + m/q_k, s_k, j_k\} \\ &\times P\{q_{k,i-1} = l/q_{k,0} = q_k\}.\end{aligned}$$

4 Numerical Results

In this section, some numerical results are presented to illustrate the impact of network and source time scales on the performance of the ABR source. While a large propagation delay is expected to impact negatively on the performance of the system, the opposite impact is expected from large ABR and VBR time scales.

The performance of the queuing system in terms of cell loss probability for the ABR source is illustrated in Figure 5. The loss probability is plotted as a function of p_f and the results are derived for a buffer capacity C of 40 and VBR source rates of 0.4 and 0.8 cells/slot. For this plot the $TC(1, T_k)$ transmission policy is used for the ABR source. As the propagation delay decreases (p_f increases), the amount of time that it takes for the ABR source to respond to the VBR source rate changes decreases. As a consequence, the duration of bandwidth mismatch decreases, causing a decrease in the ABR cell loss probability. As the VBR time scale decreases (p_s increases) , VBR source rate changes occur more frequently. As a result, more bandwidth mismatch cycles are initiated resulting in more losses. Figure 5 shows the ABR loss probability for p_s = 0.005,

$p_s = 0.001$, $p_s = 0.0005$, $p_s = 0.0001$ where $p_s = p_s(0) = p_s(1)$. Note that p_s is kept below 0.005 due to the requirement that it should be much less than p_f in order for the assumption of no multiple feedbacks to be reasonably accurate.

In Figures 6 and 7, the impact of the ABR time scale is illustrated. The ABR loss probability is plotted against the batch size for various values of buffer capacities. The VBR rates are kept equal to 0.4 and 0.8 cells/slot. The average distance between the ABR source and the network access node is assumed to be 10000 subframes (thus $p_f = 10^{-4}$). Note that, the ABR time scale (subframe lengths) is directly proportional to batch sizes, since subframe lengths are equal to $B_k T_k$. For example, for this particular example a batch size of 7 corresponds to subframe lengths of 14 and 42, where the corresponding fundamental subframe lengths are 2 and 6. Because of the single outstanding feedback assumption, p_s is kept much smaller than p_f ($p_s = 10^{-6}$). The ABR loss probability vs. the batch size is plotted in Figure 6 for $C = 10, 15, 20, 25, 30$. It can be clearly observed that for a fixed value of C, the loss probability initially decreases as the batch size increases. This behavior is reversed when the batch size exceeds a threshold. Thus, for a given buffer capacity C, there is an optimal batch size that minimizes the induced ABR cell losses. These observations can be made in Figure 7 where results for a larger range of batch sizes are plotted for $C = 10, 20$.

The reduced ABR losses for batch sizes greater than one may be attributed to the associated increased ABR source time scale. This positive impact prevails as long as the buffer capacity C is large enough to absorb the increased batches. When the batch size exceeds a threshold, it cannot be effectively absorbed by the fixed buffer size, resulting in losses which are not compensated for by the benefits from the increased time scale. The larger the value of C, the larger this threshold is expected to be, as it is clearly observed in Figures 6 and 7.

The impact of the employed ABR time scale (or batch size) on the system performance is expected to be affected by the propagation delays. Figures 8 and 9 illustrate the impact of the propagation delay on the performance of the system. Figure 8 is obtained for an average distance between the ABR source and the network access node equal to 1000 subframes (thus $p_f = 10^{-3}$). Figure 9 is obtained for an average distance of 100 subframes (thus $p_f = 10^{-2}$). As expected, the performance improves under all ABR time scales as the distance decreases, because the amount of bandwidth mismatch is proportional to the distance.

By comparing the plots for $C = 10$ in Figures 6, 8, 9 which have been obtained for an average propagation distance of 20000, 2000 and 200 slots, respectively, it may be concluded that the optimal batch size decreases with the propagation distance. For a propagation distance of 200 slots and $C = 10$ the optimal batch size is as low as 1. This trend may be attributed to the decreasing positive impact of a large ABR time scale (batch size) as the propagation delay decreases. This positive impact cannot compensate for the negative impact of a larger batch size, driving the optimal batch size to lower values.

The above can be more clearly observed in Figure 10 where the optimal batch size against the buffer capacity is plotted for average distances of 20000 and 2000 slots. It can be established that for large propagation distances, the optimal batch size is larger than 1 even for small buffer capacities.

For small propagation distances and small buffer capacities, better performance is achieved. Note that the optimal batch size seems to be linearly increasing with the buffer capacity.

In addition to the network and source time scales, the VBR source rates also have considerable effects on the performance. Figure 11 shows the affects of the VBR source rates on the loss probability. For a meaningful comparison, the mean VBR arrival rate is kept constant for the three cases considered. The VBR source rates for the three cases are 0.9 and 0.3, 0.8 and 0.4, 0.7 and 0.5 cells/slot. It can be clearly observed that the loss performance improves as the difference between the two rates decreases. For the case with $r_v(0) = 0.3$ and $r_v(1) = 0.9$, the subframe lengths are 2 and 11 slots. These subframe lengths correspond to ABR source rates of 0.5 and 0.091 respectively. Therefore, during a bandwidth mismatch initiated by a VBR source rate change, the total rate on the link can increase up to 1.4 causing potentially high losses. When the VBR source rate drops from 0.9 to 0.3, the total rate in the link decreases to 0.391 causing excessive underutilization. However, when VBR rates are 0.4 and 0.8, the corresponding ABR rates will be 0.5 and 0.167. During a bandwidth mismatch the total rate on the link will be at most 1.3 or at least 0.567. Compared to 1.4 and 0.391, the overutilization and underutilization in this case are decreased. Therefore, a decrease in the ABR loss probabilities occurs.

5 Conclusions

In this paper, the impact of network and source time scales on the performance of ABR applications is studied. The network time scale is defined as the transmission time of a cell. The VBR time scale is defined in terms of the rate of change of the VBR source rate. The ABR time scale is defined to be the minimum distance between consecutive blocks of consecutive cells generated by the ABR source. In order to evaluate the impact of these time scales, a system with one ABR and one VBR source is considered. (A system with multiple ABR sources also gave similar results.)

While the impact of the network time scale has been considered in detail in the past, that of other relevant time scales has not been investigated. The main contribution of this paper is the study of the impact of the VBR and ABR time scales on feedback based flow control.

The increased VBR time scale has a positive impact on the ABR loss performance. This is due to the fact that, less frequent VBR rate changes (or bandwidth mismatch cycles) occur when the VBR time scale increases, resulting in reduced ABR cell losses. Increasing the ABR time scale (or the batch size), initially decreases the loss probability. After the batch size exceeds a threshold, the situation is reversed. The optimal batch size -defined to be the one corresponding to this threshold- increases with the buffer capacity, as the capability of absorbing larger batches by the buffer increases with increased buffer space. For large propagation distances the optimal batch size is greater than one, even for small buffer capacities. As the propagation distance decreases, the optimal batch size decreases as well. This may be due to the reduced positive impact of the ABR time scale for decreased propagation delay.

The major conclusion of this work is that in addition to the network time scale, the ABR and VBR time scales may impact substantially a feedback based flow control as well. Thus the impact of *all system time scales* should be considered in order to accurately evaluate the effectiveness of a flow control algorithm. Also, based on this study a new class of transmission polices for the ABR sources is defined which may result in better system efficiency. Thus, departing from the periodic $B_k = 1$ policy, $B_k = n$ $(n \geq 1)$ policies may be more beneficial. Further study will focus on development of a new framework for ABR traffic management based on this new policy.

References

[1] ATM Forum, "Traffic Management Specification Version 4.0", *ATM Forum/95-0013R10*, Feb. 1995.

[2] T. Chen et al. , "The Available Bit Rate Service for Data in ATM Networks", *IEEE Communications Magazine*, pp. 56-71, May 1996.

[3] F. Bonomi and R. Morris, "The Rate Based Flow Control Framework for the Available Bit Rate ATM Service", *IEEE Network*, pp. 25-39, Mar./Apr. 1995.

[4] J. C. Bolet and A. Shankar, "Dynamical Behavior of Rate Based Flow Control Mechanism", *ACM Comp. Commun Rev.*, pp. 35-49, Apr. 1990.

[5] S. Liu et al., "Fairness in Closed Loop Rate Based Traffic Control Schemes", *ATM Forum/94-0387*, May 1994.

[6] H. T. Kung and R. Morris, "Credit Based Flow Control for ATM Networks", *IEEE Network*, pp. 40-48, Mar./Apr. 1995.

[7] H. T. Kung and K. Chang, "Receiver-Oriented Adaptive Buffer Allocation in Credit-Based Flow Control for ATM Networks," *INFOCOM'95*, pp. 239-252, 1995.

[8] J. F. Ren and J. W. Mark, "Design and Analysis of a Credit Based Controller for Congestion Control in B-ISDN/ATM Networks," *INFOCOM'95*, pp. 40-48, 1995.

[9] K. K. Ramakrishnan and P. Newman, "Integration of Rate Based and Credit Schemes for ATM Flow Control", *IEEE Network*, pp. 49-56, Mar./Apr. 1995.

[10] N. Yin and M. G. Hluchjy, "On Closed-Loop Rate Control for ATM Cell Relay Networks," *INFOCOM'94*, pp.99-108, Toronto, 1994.

[11] R. Beraldi and S. Morano, "Selective BECN Schemes for Congestion Control of ABR traffic in ATM LAN," *ICC'96*, pp. 503-507, 1996.

[12] A. Barnhart, "Baseline Performance Using PRCA Rate Control," *ATM Forum/94-0597*, July 1994.

[13] A. Charny et al., "Congestion Control with Explicit Rate Indications," *ICC'95*, pp. 1954-1963, Seattle, 1995.

[14] L. Roberts, "Enhanced PRCA," *ATM Forum/94-0735R1*, Aug. 1994.

[15] Y. T. Wang and B. Sengupta, "Performance Analysis of a Feedback Congestion Control Policy Under non-negligible Propagation Delay," *ACM SIGCOMM'91*, pp. 149-157, Sep. 1991.

[16] M. Abdelaziz and I. Stavrakakis, "Study of an Adaptive Rate Control Scheme Under Unequal Propagation Delays," *ICC'95*, pp.1964-1968, Seattle, 1995.

[17] R. Pazhyannur and R. Agrawal, "Feedback Based Flow Control in ATM Networks with Multiple Propagation Delays," *1996 IEEE*, pp. 585-93.

[18] M. Abdelaziz and I. Stavrakakis, "Study of the Effectiveness of Adaptive Rate Control for Slow Arrival Processes," *GLOBECOM'96*, pp. 1102-1106, London, Nov. 1996.

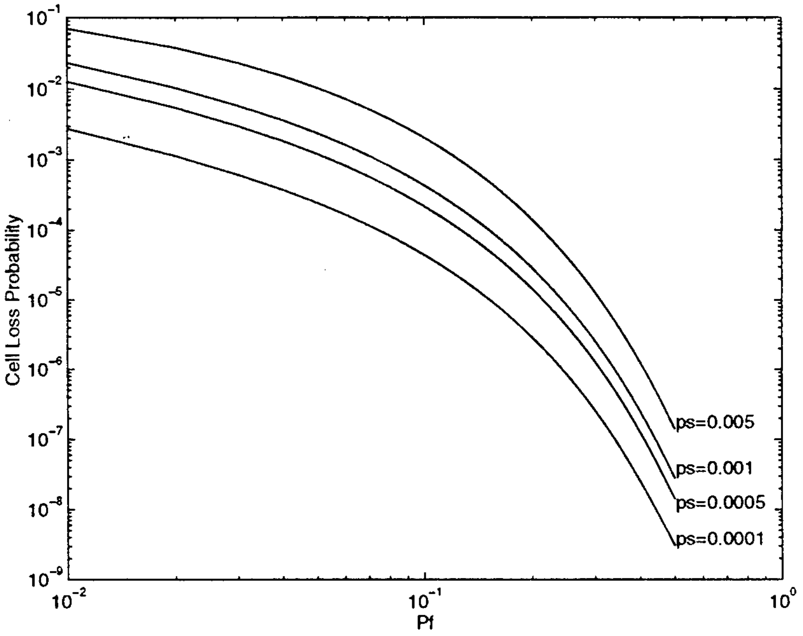


Figure 5: ABR cell loss probability vs. p_f for various p_s

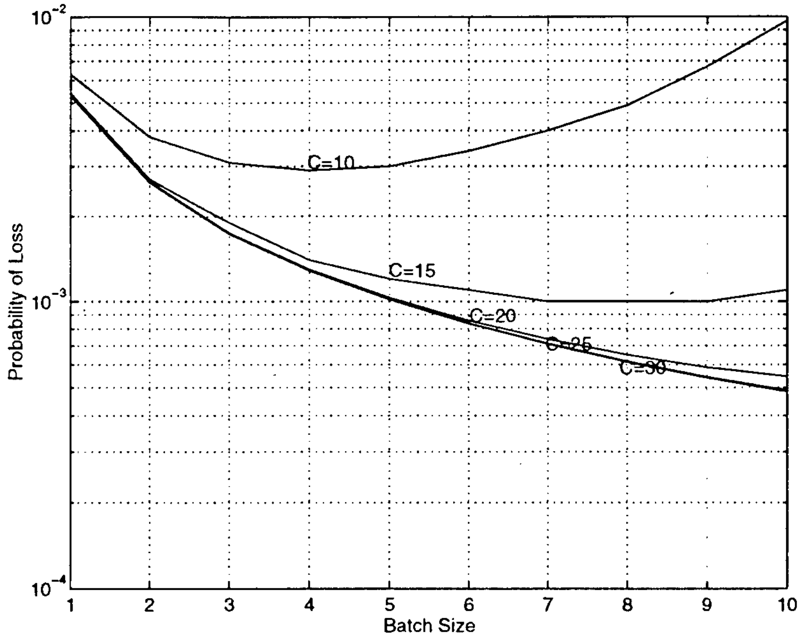


Figure 6: ABR cell loss probability vs. batch size for various values of C

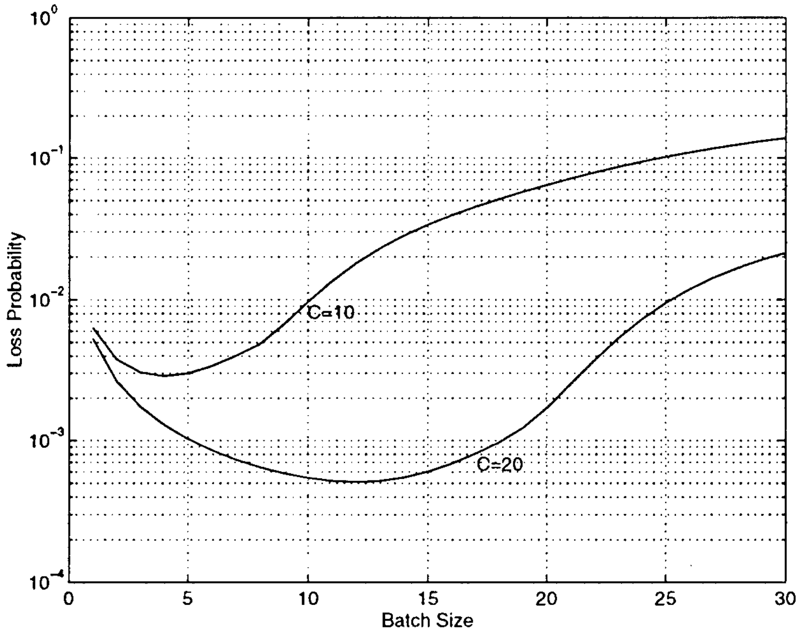


Figure 7: ABR cell loss probability vs. batch size for $C = 10$ and $C = 20$

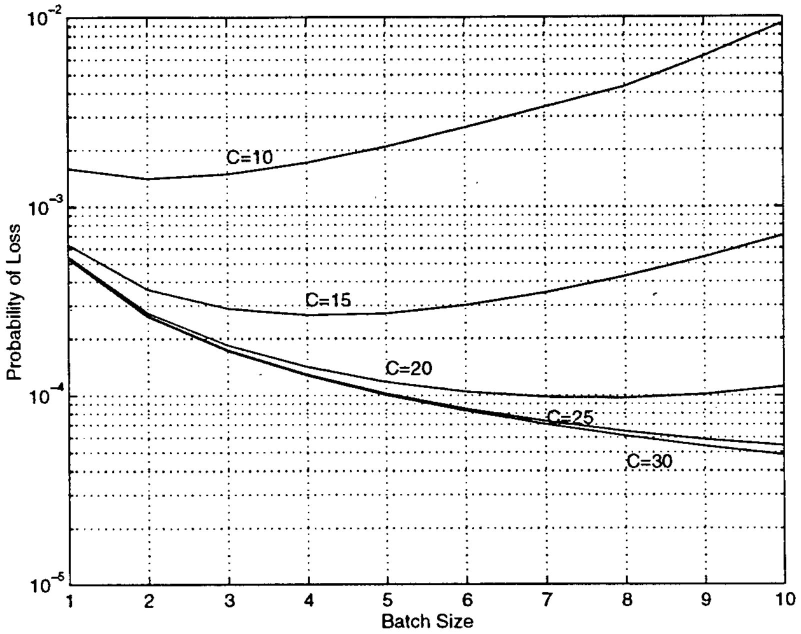


Figure 8: ABR cell loss probability vs. batch size for a distance of 2000 slots

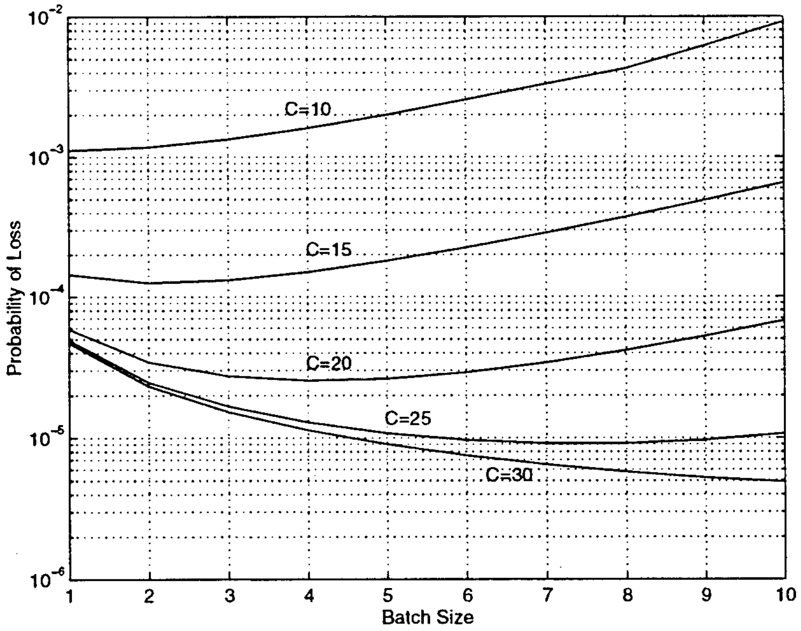


Figure 9: ABR cell loss probability vs. batch size for a distance of 200 slots

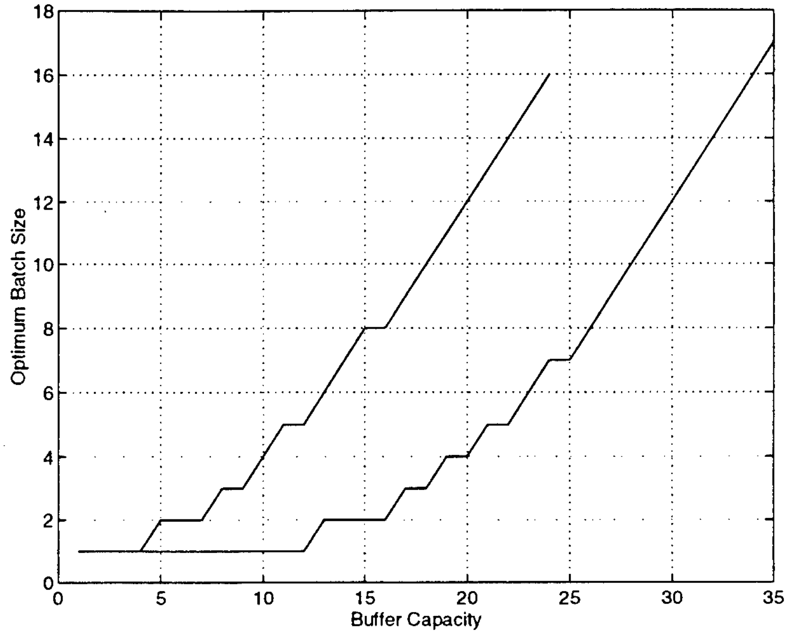


Figure 10: Optimum batch size vs. buffer capacity

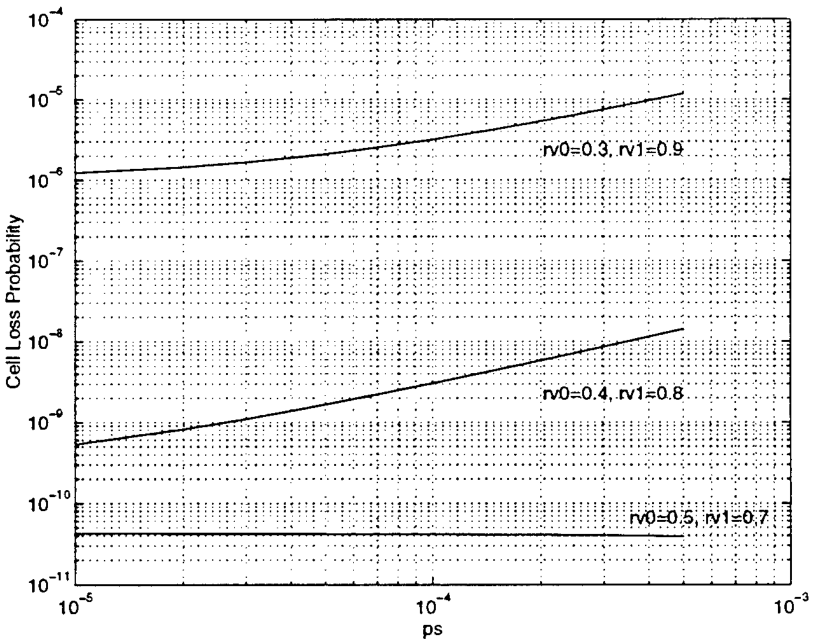


Figure 11: ABR cell loss probability vs. p_s for various VBR arrival rates

Delivering QoS Requirements to Traffic with Diverse Delay Tolerances in a TDMA Environment*

Jeffrey M. Capone   Ioannis Stavrakakis

Department of Electrical & Computer Engineering
Northeastern University
Boston, MA 02115, USA

Abstract

In *Wireless ATM networks* transmission resources are shared among geographically disperse applications with diverse Quality of Service (QoS) requirements and traffic characteristics. To provide QoS guarantees and use the bandwidth efficiently, call admission and scheduling functions are necessary. These functions should ensure the delivery of the target QoS to the supported applications while achieving statistical multiplexing gains, without explicit and continuous exchange of information between sources and scheduler.

In this ~~paper~~ research the problem of sharing resources (slots of an up-link TDMA frame) among heterogeneous Variable Bit Rate (VBR) applications with diverse QoS requirements is addressed. The QoS requirements for each application is defined in terms of a maximum tolerable packet delay and dropping probability; a packet is dropped if it experiences excess delay. The region of achievable QoS vectors is established for policies that are *work-conserving* and satisfy the *earliest due date* (EDD) service criterion (WC-EDD policies); such policies are known to optimize the overall system performance. In addition to the determination of the region of achievable QoS vectors, this study leads also to the construction of scheduling policies which deliver any performance in the region established for WC-EDD policies. Finally, an upper bound on the region of QoS vectors that can be achieved under *any* policy (not limited to the WC-EDD policies) is determined.

Please send correspondence to:
Ioannis Stavrakakis, ECE Dept. - 409 Dana Bldg., Northeastern University, Boston, MA 02115.
(E-mail: ioannis@cdsp.neu.edu)

*Research supported in part by the Advanced Research Project Agency under Grant F49620-93-1-0564 monitored by the Air Force Office of Scientific Research (AFOSR) and the National Science Foundation under Grant NCR 9628116.

Center for Communications and Digital Signal Processing
(CDSP)

TECHNICAL REPORT

TR-CDSP-97-42

Delivering QoS Requirements to Traffic with Diverse Delay Tolerances in a TDMA Environment

J.M. Capone and I. Stavrakakis

Department of Electrical & Computer Engineering
Northeastern University, Boston, MA 02115.

(email: ioannis@cdsp.neu.edu)

March 20, 1997

1 Introduction

The next-generation wireless networks are expected to offer a range of services to support integrated applications such as, voice, video. multi-media and conventional data. In addition, this new architecture should be compatible with the prevalent network architecture for integrated services over fiber/copper based channels, the Asynchronous Transfer Mode (ATM). It would only seem natural to extend the ATM protocol into the wireless environment, but this environment poses many challenges for ATM, such as an unreliable wireless link and a multi-access up-link channel. Recently, system architectures have been proposed to enable "Wireless ATM" (WATM) [1. 2. 3]. These systems employ a Data Link Control (DLC) layer, to combat the unreliability of the wireless link. and a medium access control (MAC) protocol, to organize the sharing of the multi-access channel (wireless up-link). MACs for WATM have been examined in [1, 4, 5, 6, 7] and all employ TDMA with on demand assignment of the transmission resources. Allocating resources on demand for Variable Bit Rate (VBR) sources while a call is in progress, statistical multiplexing gains are achieved. results in efficient use of the available bandwidth.

Due to the lack of resources in a wireless network. complete continuous information of the status of the distributed source packets may not be known. The information needed to assign transmissions may only be updated periodically or at discrete moments, as in [1, 4, 5. 6, 7]. These systems use a combination of control channels, piggy-backing on information bearing packets and polling procedures to communicate service request and transmission assignments between the scheduler and source. In this environment a MAC protocol should determine a transmission scheduling policy in a way that the probability of violating the maximum tolerable packet delay - which leads to packet dropping rate because of discarding packets with excess delay - does not exceed the required maximum packet dropping probability. At the same time a call admission function should ensure that the requesting service from all of the supported applications is possible to deliver. The maximum packet delay and packet dropping probability are the critical QoS parameters for real-time oriented services. The call admission and scheduling functions ensure the delivery of target QoS to the supported applications while achieving statistical multiplexing gains.

In this paper, call admission and transmission scheduling policies are considered for a TDMA system. where service requests are processed at frame boundaries. This resource structure has been widely considered in both cellular systems [8] and wireless LANs [9], as well as in recent work toward the development of wireless ATM networks [1]. Transmission scheduling policies for this environment were developed in [10] to accommodate heterogeneous Variable Bit Rate (VBR) applications with diverse packet dropping requirements. In that work, packets which are not serviced in the frame following their arrival are considered to have excess delay and are dropped. That is, all packets assume a common maximum tolerable delay of one frame. In the present work, the packets are considered to have diverse maximum delay as well as dropping probability requirements. As it becomes apparent in this paper, the added diversity regarding the maximum delay creates a number of issues. Some of these issues are addressed by considering a specific family of policies, as discussed in the main body of this paper.

In the next section, the system model considered in this work is described. In section 3, the region of achievable QoS vectors is established for policies that are *work-conserving* and satisfy the *earliest due date* (EDD) service criterion (WC-EDD policies). Determining this region is central to the call admission control problem as well as the design of effective transmission scheduling algorithms. In section 4, the probability distribution of the residual traffic process is derived for arrival processes modeled in terms of a sequence of i.i.d. random variables. Section 5 examines a class of policies that can deliver any achievable WC-EDD performance. An upper bound on the region of QoS vectors that can be achieved under *any* policy (not limited to the WC-EDD policies) is determined in section 6. The region of acceptable QoS vectors is examined in section 7. Numerical examples are presented in section 8 and the conclusions of this work are contained in section 9.

2 System Model

In this paper, the problem of sharing T resources by N heterogeneous Variable Bit Rate (VBR) source applications is considered. The source packet arrival process is described in terms of a general arrival process embedded at the boundaries of fixed length intervals called service cycles (or frames). No additional assumptions for the packet arrival process are necessary at this point. Up to T packets may be transmitted (served) during each service cycle. Depending on the Quality of Service (QoS) requirements, packets which cannot be transmitted over the service cycle following their arrival may be dropped (due to delay violation) or may be delayed to compete for service in the next frame. A TDMA system in which arrivals are considered at frame boundaries may be modeled in terms of a discrete time system in which packet delays are measured in frames (1 Frame $= L$ time units). In the system considered in this work, the N VBR sources are partitioned into two classes, $S_1 = \{1, 2, \ldots, K\}$ and $S_2 = \{K+1, K+2, \ldots, N\}$. Packets generated from sources in S_1 have a common maximum delay tolerance of L time units (1 frame) and packets generated from sources in S_2 have a common maximum delay tolerance of $2L$ time units (2 frames).

Let $\lambda_i(n)$, $i \in \{S_1 \cup S_2\}$ denote the number of newly generated packets requesting service from sources in S_1 and S_2 at the n^{th} frame boundary. The aggregate traffic from sources in S_1 is given by $\lambda_{S_1}(n) = \sum_{i \in S_1} \lambda_i(n)$, and has a delay tolerance of L time units, thus, must be either serviced or dropped over frame n. Newly generated requests from sources in S_2, $\lambda_{S_2}(n) = \sum_{i \in S_2} \lambda_i(n)$, may be either serviced or delayed to the next frame to compete for service. Packets from S_2 that have been delayed (and must be serviced or dropped in the current frame) are denoted as, $\lambda_i^{r,f}(n)$; the superscript r is used to indicate residual traffic (those packets from $\lambda_i(n-1)$, $i \in S_2$ not serviced) and the superscript f indicates its dependency on the service policy f. Thus, the total residual traffic from sources in class S_2 requiring service in frame n is given by, $\lambda^{r,f}_{S_2}(n) = \sum_{i \in S_2} \lambda^{r,f}_i(n)$.

The number of packets from source i that are dropped under policy f during frame n are given by,

$$\begin{aligned} d^f_i(n) &= \lambda_i(n) - a^f_i(n), \qquad i \in S_1 = \{1, 2, \ldots, K\}, && (1) \\ d^f_i(n) &= \lambda^{r,f}_i(n) - a^f_{i,1}(n), \quad i \in S_2 = \{K+1, K+2, \ldots, N\}; && (2) \end{aligned}$$

$a^f_i(n)$ denotes the number of packets from source i serviced during frame n under policy f; $a^f_{i,1}(n)$, $i \in S_2$, denotes the amount of service (in slots) provided to the residual traffic associated with source i during frame n. Since in any frame two types of packet from a source in S_2 may be present (new arrivals and residual traffic), the total amount of resources devoted to source $i \in S_2$ in frame n, is given by,

$$a^f_i(n) = a^f_{i,1}(n) + a^f_{i,2}(n), \qquad i \in S_2, \tag{3}$$

where $a^f_{i,2}(n)$ is the amount of available resources allocated to the new arrivals from S_2 in frame n under some policy f. Since the residual traffic in frame n consist of the new arrivals from sources in S_2 which did not receive service in frame $(n-1)$, the residual traffic in frame n under policy f is given by,

$$\lambda^{r,f}_i(n) = \lambda_i(n-1) - a^f_{i,2}(n-1) \qquad i \in S_2. \tag{4}$$

Let $d^f_i = E\left[d^f_i(n)\right]$, $a^f_i = E\left[a^f_i(n)\right]$ and $\lambda_i = E\left[\lambda_i(n)\right]$ be the (assumed time invariant) expected values of the associated quantities. The residual traffic is illustrated in the realization depicted in Fig. 1.

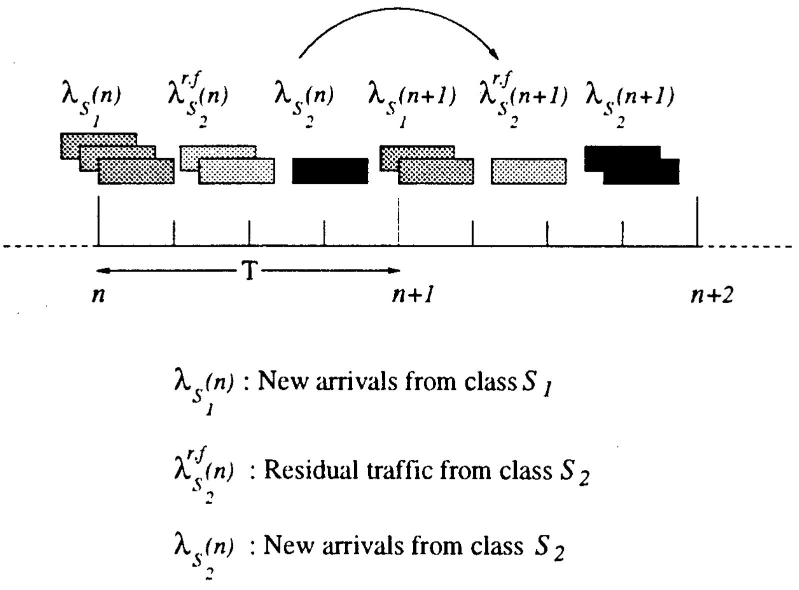

$\lambda_{S_1}(n)$
$\lambda^{r,f}_{S_2}(n)$
$\lambda_{S_2}(n)$
$\lambda_{S_1}(n+1)$
$\lambda^{r,f}_{S_2}(n+1)$
$\lambda_{S_2}(n+1)$
T
n
n+1
n+2
$\lambda_{S_1}(n)$: New arrivals from class S_1
$\lambda^{r,f}_{S_2}(n)$: Residual traffic from class S_2
$\lambda_{S_2}(n)$: New arrivals from class S_2

Figure 1: Realization of residual traffic.

If a Work Conserving Earliest Due Date (WC-EDD) policy is employed, then four out of the five packets with a service deadline (or due date) in the current frame will be served, the remaining one will be dropped, and the new arrival from class S_2 will form the residual traffic in the next frame (frame $(n+1)$). According to the WC-EDD policy, all three packets with service deadline in frame $(n+1)$ will be served during the frame, as well as one of the two packets with service deadline in frame $(n+2)$.

From the above example it is evident that the employed EDD policy imposes restrictions on the level of QoS diversification that could be achieved otherwise. For instance, new arrivals from class S_2 cannot be serviced in the presence of packets from class S_1, imposing a limit on the minimum dropping rate for sources in class S_2. This limit is higher than the dropping rate achieved if, for instance, all packets (new and residual) from sources in class S_2 had service priority over those in S_1.

In most of this paper, the class of WC-EDD polices is considered for the following reasons. First, the WC-EDD policies are known to minimize the *system* packet dropping (delay violation) probability [11], resulting in throughput maximization. Unlike a more general case in which the service deadlines or due dates would form a continuum or may be drawn from a large collection of values, only two service deadlines are considered in the TDMA environment in this work. As a consequence, a potentially large number of packets from different sources will have identical service deadlines or due dates (one of two values) and, thus, significant room for dropping rate diversification may be possible without departing from the WC-EDD policies. In addition, it is possible to determine the region of achievable QoS vectors under *any* WC-EDD policy, as well as scheduling policies delivering any QoS vector in this region.

If the QoS vector is not in the region of achievable QoS vectors under the WC-EDD policies, it can be concluded that such level of QoS diversification *may* be achieved only at the expense of system throughput[12]. This may suggest that the sharing of the resources by such diverse applications may need to be restricted by allowing for resource sharing by less diverse applications. In any case, by deriving an upper bound of the region of achievable QoS vectors under *any* WC policy, it can be determined whether a given QoS vector is not achievable.

In this work the QoS requirements of application i is described in terms of a maximum tolerable delay and a maximum dropping probability p_i; this is the probability that a packet from source i experiences a delay greater than its maximum tolerable delay and, thus, is dropped. The corresponding packet dropping rate or delay violation rate, d_i (measured in expected number of dropped packets per frame) is easily determined by, $d_i = p_i\lambda_i$, $0 \leq i \leq N$. In the rest of this paper the QoS vector associated with the supported applications will be described in terms of the dropping rates, with the understanding that these rates are induced due to violation of diverse delay tolerances. The QoS vector associated with the supported applications can be defined in terms of the (performance) packet dropping rate vector **d**,

$$\mathbf{d} = (d_1, d_2, \ldots, K, K+1, \ldots, d_N). \tag{5}$$

The first question addressed in the sequel (section 3) is whether a given QoS vectors **d** is achievable under any WC-EDD policy f. Necessary and sufficient conditions are derived in order for the QoS vector to be achievable under these policies, leading to the precise determination of the region of achievable QoS vectors **d**. The second question, addressed in section 5, is concerned with the design of scheduling policies that deliver any achievable QoS vector **d**. Also, an upper bound of the region of QoS vectors achieved under *any* WC policy (not necessarily a WC-EDD policy) is determined in section 6.

3 Region of Achievable QoS Vectors Under WC-EDD Policies

3.1 Conservation Law and Inequality Constraints

A formal definition of a work-conserving scheduling policy for the system described in the previous section is given first.

Definition 3.1 *A scheduling policy f, is work-conserving iff it satisfies the following conditions:*

$$\sum_{i=1}^{K} d_i^f(n) + \sum_{i=K+1}^{N} \lambda_i^{r,f}(n+1) = 0 \quad \text{if } \lambda_{S_1}(n) + \lambda_{S_2}^{r,f}(n) + \lambda_{S_2}(n) \leq T \tag{6}$$

$$\sum_{i=1}^{N} a_i^f(n) = T \quad \text{if } \lambda_{S_1}(n) + \lambda_{S_2}^{r,f}(n) + \lambda_{S_2}(n) > T \ . \ \square \tag{7}$$

A work-conserving policy does not waste resources (slots) as long as there is work to perform (packets to transmit). Let $S = \{S_1 \cup S_2\}$ be the set of all sources and $d_{\{S\}}^f$ denote the average system packet dropping rate under scheduling policy f, given by,

$$d_{\{S\}}^f \triangleq E\left[\sum_{i=1}^{N} d_i^f(n)\right] = \sum_{i=1}^{N} E\left[d_i^f(n)\right] = \sum_{i=1}^{N} d_i^f. \tag{8}$$

Definition 3.2 *Let $\mathcal{F}$ be the family of Work-Conserving Earliest Due Date (WC-EDD) policies.* □

Definition 3.3 *Frame n is said to be under-loaded when $\lambda_{S_1}(n) + \lambda_{S_2}^{r,f}(n) \leq T$ and overloaded when $\lambda_{S_1}(n) + \lambda_{S_2}^{r,f}(n) > T$.* □

Notice that during an overloaded frame, packets will be dropped and none of the $\lambda_{S_2}(n)$ packets will receive service under any policy $f \in \mathcal{F}$. Moreover, for any $f \in \mathcal{F}$ the following hold:

$$d_i^f(n) = \begin{cases} = 0 & \text{if } \lambda_{S_1}(n) + \lambda_{S_2}^{r,f}(n) \leq T \\ \geq 0 & \text{if } \lambda_{S_1}(n) + \lambda_{S_2}^{r,f}(n) > T \end{cases}, \ \forall i \in S, \tag{9}$$

and,

$$\lambda_i^{r,f}(n+1) = \begin{cases} \lambda_i(n) + \lambda_i^{r,f}(n) - a_i^f(n) & \text{if } \lambda_{S_1}(n) + \lambda_{S_2}^{r,f}(n) \le T \\ \lambda_i(n) & \text{if } \lambda_{S_1}(n) + \lambda_{S_2}^{r,f}(n) > T \end{cases}, \ \forall i \in S_2. \tag{10}$$

Theorem 3.1 *The system dropping rate, $d^f_{\{S\}}$, is conserved under any $f \in \mathcal{F}$ and is a lower bound on the system dropping rate induced under any policy. More specifically,*

$$\begin{aligned} d^f_{\{S\}} &\triangleq \left\{ E\left[\lambda_{S_1}(n) + \lambda_{S_2}^{r,f}(n) \;\middle|\; \lambda_{S_1}(n) + \lambda_{S_2}^{r,f}(n) > T \right] - T \right\} P\left(\lambda_{S_1}(n) + \lambda_{S_2}^{r,f}(n) > T \right) \\ &\triangleq b_{\{S\}} \quad \forall f \in \mathcal{F}. \ \square \end{aligned} \tag{11}$$

Proof. First, the first equality in (11) is proved. Summing (1) and (2) over all $i \in S$, the following is obtained:

$$\sum_{i \in S} d_i^f(n) = \sum_{i \in S_1} \lambda_i(n) + \sum_{i \in S_2} \lambda_i^{r,f}(n) - \sum_{i \in S_1} a_i^f(n) - \sum_{i \in S_2} a_{i,1}^f(n). \tag{12}$$

Since f is a WC-EDD policy it satisfies (6) and (7) of Definition 3.1 and (9) and (10) of Definition 3.2. Thus, the above expression becomes.

$$\sum_{i \in S} d_i^f(n) = \begin{cases} 0 & \lambda_{S_1}(n) + \lambda_{S_2}^{r,f}(n) \le T \\ \lambda_{S_1}(n) + \lambda_{S_2}^{r,f}(n) - T & \lambda_{S_1}(n) + \lambda_{S_2}^{r,f}(n) > T. \end{cases} \tag{13}$$

By applying the expectation operator in (13), the first equality is obtained. To prove the second equality in (11) - and thus, show that d^f_S is conserved - it suffices to show that $\lambda_{S_2}^{r,f}(n)$ is independent of the policy $f \in \mathcal{F}$. For any $f \in \mathcal{F}$,

$$\lambda_{S_2}^{r,f}(n+1) = \begin{cases} \max\{0, \lambda_{S_2}(n) + \lambda_{S_1}(n) + \lambda_{S_2}^{r,f}(n) - T\} & \text{if } \lambda_{S_1}(n) + \lambda_{S_2}^{r,f}(n) \le T \\ \lambda_{S_2}(n) & \text{if } \lambda_{S_1}(n) + \lambda_{S_2}^{r,f}(n) > T. \end{cases} \tag{14}$$

Therefore, $\lambda_{S_2}^{r,f}(n)$ depends on $\lambda_{S_2}^{r,f}(n-1)$ which may in-turn be dependent on f and is otherwise independent from f. By induction $\lambda_{S_2}^{r,f}(n)$ is only dependent on the initial conditions $\lambda_{S_2}^{r}(0)$ which is independent of the policy and equal to zero. Thus, the *total* residual traffic in frame n is independent of the policy $f \in \mathcal{F}$ and therefore can be denoted as $\lambda_{S_2}^{r}(n)$. That is,

$$\lambda_{S_2}^{r,f}(n) = \sum_{i \in S_2} \lambda_i^{r,f}(n) \triangleq \lambda_{S_2}^{r}(n), \quad \forall n, \quad \forall f \in \mathcal{F}, \tag{15}$$

and therefore proving the second equality. Finally, it is a well known result that $d^f_{\{S\}}$ is the minimum dropping rate since it is induced by a WC-EDD policy[11]. □

Since the total residual traffic in frame n, $\lambda_{S_2}^{r}(n)$, is independent of the policy $f \in \mathcal{F}$, the following corollary is self-evident in view of the previous theorem.

Corollary 3.1 *The number of service opportunities for the new arrivals from S_2 in frame n is independent of the selected $f \in \mathcal{F}$. Therefore, the residual traffic for any source i in frame $(n+1)$, $\lambda_i^{r,f}(n+1)$, is only dependent on the policy f chosen in the present frame, n, and not in past frames.*

Corollary 3.1 is employed in the proof of Theorem 3.5 and in the development of a class of policies in section 5.

Let $\lambda_g(n) = \sum_{i \in g} \lambda_i(n)$ be the aggregate arrivals from sources in subset g in frame n. Let d_g^f denote the aggregate packet dropping rate associated with sources in group g only, under policy f; all N sources in S are assumed to be present and served under the policy f. d_g^f is defined by,

$$d_{\{g\}}^f = E\left[\sum_{i \in g} d_i^f(n)\right] = \sum_{i \in g} E\left[d_i^f(n)\right] = \sum_{i \in g} d_i^f, \quad g \subseteq S. \tag{16}$$

The following lemma will be used in the proof of the theorem that follows.

Lemma 3.1 *Let $g \subseteq S_2$ and let $\lambda_g^{r,f}(n) = \sum_{i \in g} \lambda_i^{r,f}(n)$ be the aggregate residual traffic form subset g under some policy f. Then,*

$$\lambda_g^{r,f}(n) \geq \lambda_g^r(n), \quad \forall n \quad \textit{and} \quad \forall f \in \mathcal{F}, \tag{17}$$

where $\lambda_g^r(n)$ is given by,

$$\lambda_g^r(n+1) = \begin{cases} \max\{0, \lambda_g(n) + \lambda_{S_1}(n) + \lambda_{S_2}^r(n) - T\} & \textit{if } \lambda_{S_1}(n) + \lambda_{S_2}^r(n) \leq T \\ \lambda_g(n) & \textit{if } \lambda_{S_1}(n) + \lambda_{S_2}^{r,f}(n) > T. \end{cases} \tag{18}$$

$\lambda_g^r(n)$ is called the minimum residual traffic process for sources in $g \in S_2$, for a system that is served under a WC-EDD policy. □

Proof. The above inequality is self-evident since $\lambda_g^r(n)$ corresponds to the residual traffic under a policy in which new arrivals from sources in g are given service priority new arrivals in $\{S_2 - g\}$. □.

Theorem 3.2 *Let $b_{\{g\}}$ denote the lower bound for the aggregate packet dropping rate for sources in set g, $g \subset S$, under any policy $f \in \mathcal{F}$. Then this bound is given by,*

$$b_{\{g\}} = \left\{E\left[\lambda_g^r(n) \;\middle|\; \lambda_g^r(n) > T\right] - T\right\} P\left(\lambda_g^r(n) > T\right), \; g \subseteq S_2. \tag{19}$$

$$b_{\{g\}} = \left\{E\left[\lambda_g(n) \;\middle|\; \lambda_g(n) > T\right] - T\right\} P\left(\lambda_g(n) > T\right), \; g \subseteq S_1. \tag{20}$$

$$b_{\{g\}} = \left\{E\left[\lambda_x(n) + \lambda_y^r(n) \;\middle|\; \lambda_x(n) + \lambda_y^r(n) > T\right] - T\right\} P\left(\lambda_x(n) + \lambda_y^r(n) > T\right), \tag{21}$$

$$g =\subseteq S,$$

$$x = \{g \cap S_1\} \neq \emptyset \textit{ and } y = \{g \cap S_2\} \neq \emptyset$$

Proof. If $g \subseteq S_2$, the bound $b_{\{g\}}$ is achieved when $\lambda_g^r(n)$ packets do not see any interference from packets in S_1 and $\{S_2 - g\}$, or equivalently, all T resources are available to service the $\lambda_g^r(n)$ packets. Therefore, by employing $\lambda_g^r(n)$ in (11) the proof follows. The probability distribution of the minimum residual traffic, $\lambda_g^r(n)$, is calculated from (18) as in section 4. If $g \subseteq S_1$, the proof can be derived as in [10]. If $g \subset S$, where $x = \{g \cap S_1\}$ and $y = \{g \cap S_2\}$ are non-empty, $b_{\{g\}}$ is achieved when the traffic $\lambda_x(n) + \lambda_y^r(n)$ does not see interference from traffic from sources in $\{S - g\}$. By generalizing (19), the proof follows. □

The following class of policies is needed for the establishment of the region of achievable QoS vectors under the WC-EDD policies, as well as for the development of policies which deliver any achievable QoS vector for this region.

Definition 3.4 *A Deadline-Sensitive Ordered Head of Line (DSO-HoL) priority service policy is defined to be the policy which first separates packets into two sets: packets having a service deadline in the current frame and packets having a service deadline in the next frame. Packets from sources having a service deadline in the current frame are serviced according to a priority service policy,* $\pi = (\pi_1, \pi_2, \ldots, \pi_N)$*:* $\pi_i \in \{1, 2, \ldots, N\}$, $\pi_i \neq \pi_j$, $1 \leq i,\ j \leq N$. *The index of* π_i *indicates the order of the priority given to the* π_i *source to service packets having a service deadline in the current frame. None of the* π_j *sources,* $j > i$*, may be served as long as packets with current service deadline from sources* π_k, $k \leq i$*, are present. After servicing the packets having a current deadline, the same service policy,* $\pi = (\pi_1, \pi_2, \ldots, \pi_N)$*, is followed for packets from sources that are present and do not have a current deadline.* □

Theorem 3.3 *The following constraints associated with the induced packet dropping rate vector* $\mathbf{d}^f = (d_1^f, d_2^f, \ldots, d_N^f)$ *are satisfied by any scheduling policy* $f \in \mathcal{F}$,

$$d_{\{g\}}^f \geq b_{\{g\}}, \quad \forall g \subset S, \quad \forall f \in \mathcal{F} \tag{22}$$

where $d_{\{g\}}^f$ *and* $b_{\{g\}}$ *are given in (16) and (19)-(21).* □

Proof. Let $g \subset S$ and $x = g \cap S_1$ and $y = g \cap S_2$. Suppose that the packets from sources in g are served under a scheduling policy $f_o \in \mathcal{F}$ according to which they are given Deadline Sensitive Head of the Line (DS-HoL) priority over the packets from sources in $\{S - g\}$. That is, during overload conditions no packet from sources in $\{S - g\}$ is served unless no packets with a current service deadline from sources in g are present. During under-load conditions, no packets from sources in $\{S_2 - y\}$ are serviced while packets from sources in $y = \{g \cap S_2\}$ are present, and the minimum residual traffic is achieved for set y. Thus,

$$\sum_{i \in g} d_i^{f_o}(n) = \begin{cases} 0 & \text{if } \lambda_x(n) + \lambda_y^r(n) \leq T \\ \lambda_x(n) + \lambda_y^r(n) - T & \text{if } \lambda_x(n) + \lambda_y^r(n) > T. \end{cases} \tag{23}$$

From (21) and (23) it is clear that,

$$d^{f_o}_{\{g\}} = b_{\{g\}}, \tag{24}$$

for all policies $f_o \in \mathcal{F}$ that provide DS-HoL priority to the packets from sources in g. Since no other policy $f \in \mathcal{F}$ can provide better service (that is, lower aggregate packet dropping rate) to sources in g than f_o, it is evident that,

$$d^{f}_{\{g\}} \geq b_{\{g\}}, \ \ \forall f \in \mathcal{F}, \ \ g \subset S. \ \Box \tag{25}$$

3.2 Region of Achievable QoS Vectors

The main result of this section is the determination of the region $\mathcal{D}$ of the achievable QoS vectors $\mathbf{d}$ under WC-EDD policies. The following corollary provides a set of necessary conditions in order for a QoS vector $\mathbf{d}$ to be achievable, followed by a corollary regarding an upper bound on the achievable region for QoS vectors $\mathbf{d}$ under WC-EDD policies. Their proofs are self-evident in view of Theorems 3.1 and 3.3.

Corollary 3.2 *A necessary condition in order for a QoS vector* $\mathbf{d} = (d_1, d_2, \ldots, d_N)$ *to be achieved by a policy* $f \in \mathcal{F}$ *is that its components satisfy the following constraints,*

$$\begin{aligned} d_{\{g\}} &\geq b_{\{g\}} \quad \forall g \subset S & (26) \\ d_{\{S\}} &= b_{\{S\}}. \ \Box & (27) \end{aligned}$$

Corollary 3.3 *Let* $\mathcal{D}^u$ *denote the collection of all vectors* $\mathbf{d}$ *satisfying (26) and (27). Then* $\mathcal{D}^u$ *is an upper bound on the region* $\mathcal{D}$ *of achievable QoS vectors* $\mathbf{d}$. *That is,*

$$\mathcal{D} \subseteq \mathcal{D}^u. \ \Box \tag{28}$$

The following theorem describes the vectors contained in $\mathcal{D}^u$.

Theorem 3.4 *Any vector in the set* $\mathcal{D}^u$ *can be expressed as a convex combination of extreme points (vertices) of* $\mathcal{D}^u$*; that is,* $\mathcal{D}^u$ *may be expressed as the convex hull of its extreme points,* $\mathcal{D}^u$*=conv[ext(*$\mathcal{D}^u$*)].* □

Proof. The proof follows from the fact that $\mathcal{D}^u$ is a bounded set defined by a finite intersection of closed half spaces, (see (26) and (27)). Then by definition, $\mathcal{D}^u$ is a polytope [13] and Theorem 3.4 follows directly from properties of polytopes [13]. □

The following theorem establishes a relationship between scheduling policies and the vertices of $\mathcal{D}^u$.

Theorem 3.5 $\mathbf{d}^*$ *is a vertex (extreme point) of the set* $\mathcal{D}^u$ *iff* $\mathbf{d}^*$ *is a dropping rate vector resulting from a Deadline-Sensitive Ordered HoL (DSO-HoL) priority service policy,* $\boldsymbol{\pi}= (\pi_1, \pi_2, \ldots, \pi_N)$. □

Proof. Assume that $\mathbf{d}^*$ is a vertex of $\mathcal{D}^u$. Then $\mathbf{d}^*$ must lie at the intersection of N hyper-planes and its coordinates must satisfy N simultaneous, linearly independent equations (definition of a polytope vertex, see [13]), given by,

$$\sum_{i \in g_j} d_i^* = b_{\{g_j\}}, \quad j = 1, 2, \ldots, N, \tag{29}$$

where one of the g_j's is the set $S = \{1, 2, \ldots, N\}$ and the remaining $(N-1)$ are proper, non-empty and different subsets of S. Lemma 10.3 (see appendix) establishes that the subsets g_i's are strictly included in each other. Therefore, by adopting the order $g_1 \subset g_2 \subset \ldots \subset g_{N-1} \subset g_N \equiv S$, the g_i's are given by,

$$\begin{aligned} g_1 &= \{\pi_1\} \\ g_1 \subset g_2 &= \{\pi_1, \pi_2\} \\ g_2 \subset g_3 &= \{\pi_1, \pi_2, \pi_3\} \\ &\vdots \\ g_{N-1} \subset g_N &= \{\pi_1, \pi_2, \ldots, \pi_N\} = \{1.2.\ldots, N\}, \end{aligned} \tag{30}$$

where $\pi_i \in \{1.2.\ldots, N\}$, $\pi_i \neq \pi_j$, for $i \neq j$, $1 \leq i.\ j \leq N$. By using (21) the following can be obtained from (29).

$$\begin{aligned} d^*_{\{\pi_1\}} &= b_{\{\pi_1\}} \\ d^*_{\{\pi_2\}} &= b_{\{\pi_1,\pi_2\}} - b_{\{\pi_1\}} \\ d^*_{\{\pi_3\}} &= b_{\{\pi_1,\pi_2.\pi_3\}} - b_{\{\pi_1,\pi_2\}} \\ &\vdots \\ d^*_{\{\pi_N\}} &= b_{\{\pi_1,\pi_2,\ldots,\pi_N\}} - b_{\{\pi_1,\pi_2,\ldots,\pi_{N-1}\}}, \end{aligned} \tag{31}$$

which is precisely the dropping rate vector induced by the DSO-HoL priority policy $\boldsymbol{\pi}= (\pi_1,\pi_2,\ldots,\pi_N)$. Thus, for any vertex (extreme point) there exists a DSO-HoL priority policy that induces it. Since it is easy to see that the dropping rate vector resulting from a DSO-HoL priority policy must satisfy (31), it is evident that these dropping rate vectors will be the vertices of $\mathcal{D}^u$. □

Fig. 2 and Fig. 3 provide a graphical illustration of the region $\mathcal{D}^u$ for the case of $N = 2$ and $N = 3$ sources, respectively. The extreme points $\mathbf{d}_{ext-i}$'s correspond to QoS vectors induced by the $(N!)$ DSO-HoL priority policies $\boldsymbol{\pi}^i= (\pi_1, \pi_2, \ldots, \pi_N)$, $1 \leq i \leq N!$, as shown in Theorem 3.5. Referring to Fig. 2, it may be observed that the policy $(\pi_1, \pi_2) = (1, 2)$ corresponds to the intersection of the line for the lower bound on the packet dropping rate line for source 1, $b_{\{1\}}$, with the system dropping rate line $b_{\{1,2\}}$. Similarly, the second extreme point induced by the policy

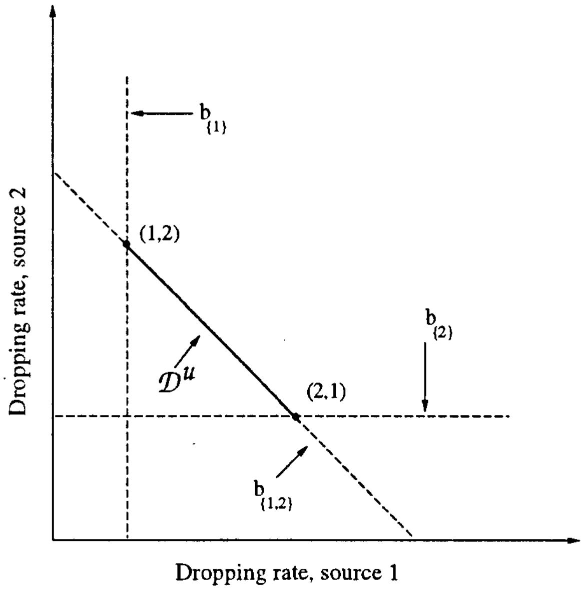


Figure 2: The region (polytope) $\mathcal{D}^u$ for a system with two sources.

$(\pi_1, \pi_2) = (2, 1)$ is the intersection of the lower bound on the packet dropping rate for source 2, $b_{\{2\}}$ and $b_{\{1,2\}}$. Similar observations can be made for the region $\mathcal{D}^u$ for a system of $N = 3$ sources shown in Fig. 3.

Let $\mathcal{D}$, $\mathcal{D} \subseteq \mathcal{D}^u$, be the achievable region of QoS vectors $\mathbf{d}$. The following theorem establishes its convexity.

Theorem 3.6 *Let* $\mathbf{d}_1, \mathbf{d}_2 \in \mathcal{D}$, *then* $\mathbf{d}_3$, *where*

$$\mathbf{d}_3 = \alpha \mathbf{d}_1 + (1 - \alpha)\mathbf{d}_2, \qquad \alpha \geq 0, \quad \alpha \leq 1, \tag{32}$$

is also in $\mathcal{D}$. *That is* $\mathcal{D}$ *is convex.* □

Proof. Let $\mathbf{d}_1$ ($\mathbf{d}_2$) be the dropping rate vector induced by the policy f_1 (f_2) $\in \mathcal{F}$; that is, $\mathbf{d}_1, \mathbf{d}_2 \in \mathcal{D}$. Consider a scheduling policy f_3 that at each under-loaded frame decides to follow the scheduling rule of policy f_1 with probability α and policy f_2 with probability $(1 - \alpha)$; the decisions over consecutive under-loaded frames are independent. Notice that no dropping occurs during under-loaded frames, and thus, any policy $f \in \mathcal{F}$ could be selected. However, the residual traffic in the first overloaded frame depends on the policy selected in the previous frame (only), the policy selection in under-loaded frames does affect the performance (drops) during overloaded frames[1]. Decisions over the first overloaded frame following an under-loaded one, follow the policy

[1] In fact any policy $f \in \mathcal{F}$ can be selected during an under-loaded frame, except for the frame before an overloaded one, since shown in corollary 3.1, the residual traffic for source i in frame n, $\lambda_i^{r,f_{1(2)}}(n)$ is only dependent on the policy $f_{1(2)}$ selected in frame $(n-1)$ and not in earlier frames.

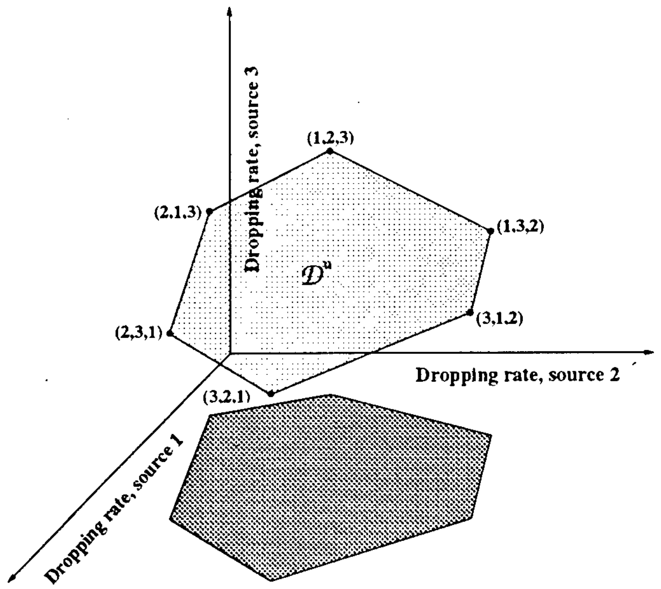


Figure 3: The region (polytope) $\mathcal{D}^u$ for a system with three sources.

selected for the previous frame, ensuring that the $\lambda_i^{r,f_{1(2)}}(n)$ packets are serviced under policy $f_{1(2)}$[2]. Over overloaded frames other that the first of an overloaded period, policy $f_1(f_2)$ is selected with probability α $(1-\alpha)$ independently from the selection in the previous frame. Note that the selected policy during an overloaded frame does *not* affect the residual traffic in the next frame, refer to (18). It should also be noted that the aggregate traffic, $\lambda_{S_1}(n)+\lambda^r_{S_2}(n)$, is independent of the policy $f \in \mathcal{F}$ (Theorem 3.1). This implies that the occurrence of overloaded periods is independent of the policy and the length of consecutive overload periods are independent and identically distribute random variables, regardless of the selected policy. In view of the above, it is easy to establish that the dropping rate performance of policy $f_{1(2)}$ is induced with probability α $(1-\alpha)$. Thus, the packet dropping rate induced by policy S_3 is given by,

$$\mathbf{d}_3 = \alpha\mathbf{d}_1 + (1-\alpha)\mathbf{d}_2. \tag{33}$$

Since $\mathbf{d}_3$ is achieved by a policy in $\mathcal{F}$, $\mathbf{d}_3 \in \mathcal{D}$, establishing the convexity of $\mathcal{D}$. □

The next theorem establishes the region $\mathcal{D}$ of achievable QoS vectors.

Theorem 3.7 $\mathcal{D} \equiv \mathcal{D}^u$. □

Proof. Since $\mathcal{D} \subseteq \mathcal{D}^u$ (Corollary 3.3), it suffices to establish that $\mathcal{D}^u \subseteq \mathcal{D}$ to complete the proof. Notice that $\mathcal{D}^u$=*conv*[*ext*($\mathcal{D}^u$)] (Theorem 3.4) and that $\mathcal{D}$ is convex (Theorem 3.6) and, thus, if $\{ext(\mathcal{D}^u)\} \subseteq \mathcal{D}$ then $\mathcal{D}^u \subseteq \mathcal{D}$. The latter holds true since the extreme points of $\mathcal{D}^u$ are induced

[2] Otherwise, packets dropped in the overloaded frame under policy f_1 would be associated with the residual traffic generated in the previous (under-loaded) frame under policy f_2; the resulting performance would be that of neither f_1 nor f_2.

by the DSO-HoL priority policies (Theorem 3.5) and, thus, these points are in the region $\mathcal{D}$ of achievable QoS vectors. □

4 Analysis of Residual Traffic

In this section the probability distribution of the minimum residual traffic process, $\lambda_g^r(n)$, is derived when the arrival processes $\lambda_i(n), i \in S$, are independent and each modeled by a sequence of independent identically distributed random variables. As seen in (14), future evolution of the total residual traffic process, $\lambda_{S_2}^r(n+1)$, depends only on the present values of $\lambda_{S_2}^r(n)$, $\lambda_{S_1}(n)$ and $\lambda_{S_2}(n)$. Therefore, under the i.i.d. assumption for the arrival processes, $\lambda_{S_2}^r(n)$ is Markovian with transition probabilities,

$$P(\lambda_{S_2}^r(n+1) = i \big| \lambda_{S_2}^r(n) = j) = \begin{cases} P(\lambda_{S_2}(n) + \min(j + \lambda_{S_1}(n), T) - T = i) & i > 0, \\ P(\lambda_{S_2}(n) + \min(j + \lambda_{S_1}(n), T) \leq T) & i = 0. \end{cases} \quad (34)$$

Assuming the stationary distribution of $\lambda_{S_2}^r(n)$ exists, the distribution vector, $\boldsymbol{\lambda}_{S_2}^r$, is easily computed as,

$$\boldsymbol{\lambda}_{S_2}^r P = \boldsymbol{\lambda}_{S_2}^r, \quad (35)$$

where P is the probability transition matrix with elements $p_{j,i}$ given in (34).

The distribution of the minimum residual traffic process for subset g in (18) is found by conditioning on $\lambda_{S_2}^r(n)$. As shown in Theorem 3.1, no matter what the subset g under consideration, the quantity $\lambda_{S_2}^r(n)$ is conserved. The stationary distribution for $\lambda_g^r(n)$ is found as,

$$P(\lambda_g^r(n) = i) = \sum_j P(\lambda_g^r(n) = i \big| \lambda_{S_2}^r(n-1) = j) P(\lambda_{S_2}^r(n-1) = j). \quad (36)$$

From the i.i.d. assumption,

$$P(\lambda_g^r(n+1) = i \big| \lambda_{S_2}^r(n) = j) = \begin{cases} P(\lambda_g(n) + \min(j + \lambda_{S_1}(n), T) - T = i) & i > 0, \\ P(\lambda_g(n) + \min(j + \lambda_{S_1}(n), T) \leq T) & i = 0. \end{cases} \quad (37)$$

5 A Class of Policies

Let $C_{DSO-HoL}$ denote the class of Delay-Sensitive Ordered Head of Line (DSO-HoL) service policies π^i introduced in section 3.1.

Definition 5.1 *In each under-loaded frame, a Mixing DSO-HoL Policy f_m decides to follow the DSO-HoL policy π^i with probability α_i. $\alpha_i \geq 0$, $1 \leq i \leq N!$, $\sum_{i=1}^{N!} \alpha_i = 1$; decisions over consecutive under-loaded or overloaded frames are independent. The DSO-HoL policy from the previous under-loaded frame is chosen in the first overloaded frame. Clearly, f_m is completely determined by the $N!$ dimensional vector $\boldsymbol{\alpha}$, $\boldsymbol{\alpha} \geq \mathbf{0}$, $\mathbf{1} \cdot \boldsymbol{\alpha} = 1$. Let $M_{DSO-HoL}$ denote the class of such policies.* □

The proof of the following theorem follows directly from the proof of Theorem 3.6.

Theorem 5.1 *The dropping rate vector induced by a Mixing DSO-HoL policy is given by,*

$$\mathbf{d}^{f_m} = \sum_{i=1}^{N!} \mathbf{d}^{f_i} \alpha_i. \tag{38}$$

The following theorem establishes the main results of this section.

Theorem 5.2 *For each packet dropping rate vector* $\mathbf{d} \in \mathcal{D}$ *there exist a policy* $f_m \in M_{DSO-HoL}$ *that induces* $\mathbf{d}$. □

Proof. Let $\mathbf{d} \in \mathcal{D}$, then $\mathbf{d} = \sum_{i=1}^{N!} \alpha_i \mathbf{d}_{ext-i}$ for some $\boldsymbol{\alpha}= (\alpha_1, \alpha_2, \ldots, \alpha_{N!})$ where $\alpha_i \geq 0$, $1 \leq i \leq N!$, $\sum_{i=1}^{N!} \alpha_i = 1$, since any point in $\mathcal{D}$ can be written as a convex combination of the extreme points (vertices) $\mathbf{d}_{ext-i}$ of $\mathcal{D}(= \mathcal{D}^u)$; each $\mathbf{d}_{ext-i}$ is induced by some policy in $C_{DSO-HoL}$ (Theorem 3.5).

Let f_m be the mixing policy which selects the HoL priority $\boldsymbol{\pi}^i$ (that induces $\mathbf{d}_{ext-i}$) with probability α_i. The packet dropping rate vector $\mathbf{d}^{f_m}$ induced by f_m is given by,

$$\mathbf{d}^{f_m} = \sum_{i=1}^{N!} \alpha_i \mathbf{d}_{ext-i} \tag{39}$$

and thus f_m induces $\mathbf{d}$. □

The following corollary is obvious in view of the above theorem.

Corollary 5.1 *Let* $\mathbf{d} \in \mathcal{D}$ *be a target packet dropping rate vector. The mixing policy* $f_m \equiv \boldsymbol{\alpha}$ *induces* $\mathbf{d}$, *where* $\boldsymbol{\alpha}$ *is such that,*

$$E_{\boldsymbol{\alpha}}[\mathbf{d}_{ext}] \triangleq \sum_{i=1}^{N!} \alpha_i \mathbf{d}_{ext-i} = \mathbf{d}, \tag{40}$$

$$\boldsymbol{\alpha} \geq \mathbf{0}, \tag{41}$$

$$\mathbf{1} \cdot \boldsymbol{\alpha} = 1, \tag{42}$$

where $E_{\boldsymbol{\alpha}}[.]$ *is weighted average of the set of extreme points,* $\mathbf{d}_{ext}$, *of* $\mathcal{D}$ *with respect to the Probability Mass Function* $\boldsymbol{\alpha}$. □

6 Upper Bound on the Region of Achievable QoS Vectors Under Any Service Policy

In the previous section the region of achievable QoS vectors induced by WC-EDD policies (denoted in this section by $\mathcal{D}^{EDD}$) was determined and described in terms of conditions (26) and (27), restated (using superscript EDD) as,

$$\begin{aligned} d_{\{g\}} &\geq b_{\{g\}}^{EDD} \quad \forall g \subset S, \\ d_{\{S\}} &= b_{\{S\}}^{EDD}; \end{aligned} \tag{43}$$

$b_{\{S\}}^{EDD}$ and $b_{\{g\}}^{EDD}$ are lower bounds on the performance of WC-EDD policies as given in (11) and (21), respectively. It is well known that the WC-EDD polices optimize the system (S) performance by minimizing the system dropping rate. Therefore, any policy that attempts to improve (decrease) the dropping rate for a subset of sources g beyond the lower bound shown in (18)-(21) by relaxing the EDD condition will result in increased system (S) system dropping rate. That is, the lower bounds on the dropping rates achieved by the WC-EDD policies and the class of policies which do not necessarily satisfy the EDD condition (denoted here as an Unconstrained (UC) policy) satisfy the following conditions.

$$\begin{aligned} b_{\{g\}}^{EDD} &\geq b_{\{g\}}^{UC} \quad \forall g \subset S, \\ b_{\{S\}}^{UC} &\geq b_{\{S\}}^{EDD}. \end{aligned} \tag{44}$$

$b_{\{g\}}^{UC}$ is the *unconstrained* lower bound on the dropping rate for the sources in set g. This bound is achieved by considering that packets from sources in set g *only* are present and serviced under a work-conserving earliest due date policy; sources in $\{S-g\}$ are considered to be absent. $b_{\{g\}}^{UC}$ is calculated by applying (11) and (14) to the set g only. That is, replacing S_1 and S_2 with $x = \{g \cap S_1\}$ and $y = \{g \cap S_2\}$ in (11) and (14), respectively.

Since no policy can do better for sources in set g than $b_{\{g\}}^{UC}$ when all sources in S are present, the following necessary conditions must be satisfied by any QoS vector $\mathbf{d}$ which is achieved under some policy.

$$\begin{aligned} d_{\{g\}} &\geq b_{\{g\}}^{UC} \quad \forall g \subset S, \\ d_{\{S\}} &\geq b_{\{S\}}^{EDD}. \end{aligned} \tag{45}$$

The following proposition is self-evident in view of the above discussion.

Proposition 6.1 *An upper bound on the region of QoS vectors achieved under <u>any</u> policy, $\mathcal{D}^{UC,u}$, is given by (45).* □

Fig. 4 depicts the region $\mathcal{D}^{EDD}$ and $\mathcal{D}^{UC,u}$ for the case of two sources.

7 Region of Acceptable QoS Vectors

Definition 7.1 *The region of acceptable QoS vectors, associated with the region of achievable QoS vectors $\mathcal{D}$, $\mathcal{A}(\mathcal{D})$, is defined to be the region of vectors $\mathbf{d}$ satisfying,*

$$\begin{aligned} d_{\{g\}} &\geq b_{\{g\}} \quad \forall g \subset S, \\ d_{\{S\}} &\geq b_{\{S\}}. \ \square \end{aligned} \tag{46}$$

Proposition 7.1 *If $\mathbf{d} \in \mathcal{A}(\mathcal{D})$ then there exists a vector $\mathbf{d}' \in \mathcal{D}$ which is such that $d'_i \leq d_i \ \forall i \in S$.* □

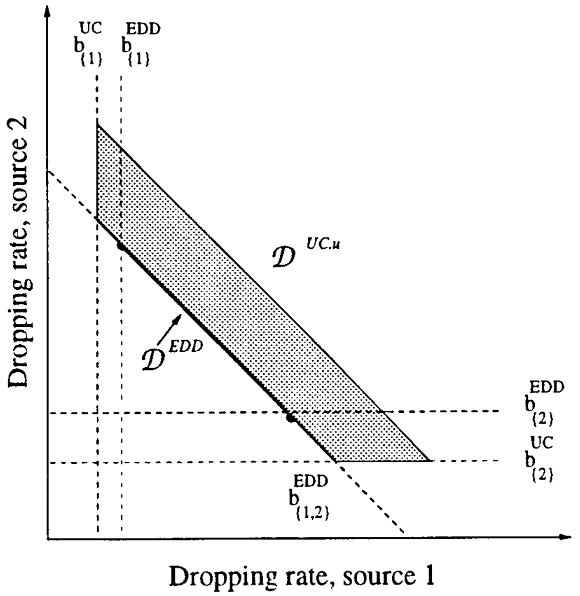


Figure 4: Necessary performance bounds under any policy and the sufficient bounds for WC-EDD family of policies.

The proof of the above proposition may be found in [10]. Proposition 7.1 implies that if the required QoS vector **d** is in $\mathcal{A}(\mathcal{D})$, there exist a policy which can deliver **d** or *better* (that is, less than the dropping rate required by any source). Fig. 5 depicts the acceptable region $\mathcal{A}(\mathcal{D}^{EDD})$ and the upper bound on the region of achievable QoS vectors $\mathcal{D}^{UC,u}$.

$\mathcal{A}(\mathcal{D}^{EDD})$ contains all the QoS vectors which can either be delivered exactly, or a better QoS vector can be delivered under some WC-EDD policy. $\mathcal{D}^{UC,u}$ is an upper bound on the QoS vectors which can be delivered under any policy.

8 Numerical Example and Verification through Simulation

In this section two examples are presented for a system with two and three sources, respectively, competing for T slots in a TDMA frame. The source packet arrival processes are assumed to be mutually independent. Each process is described in terms of a sequences of independent and identically distributed random variables embedded at the frame boundaries. Let $E^k = \{0, 1, \ldots, M^k - 1\}$ denote the state space of the arrival process associated with source k.

Consider a system with 5 ($T = 5$) slots per frame and 2 sources, $k = 2$. In this example $M^1 = 7$ and $M^2 = 7$ and when in state i, a source generates i packets in the current frame. Source 1 has a maximum arrival rate of 6 packets per frame with an average per frame arrival rate of 3.6. Source 2 has an average arrival rate of 3.2 packets per frame with a maximum arrival rate of 6 packets per frame. Sources 1 and 2 have state probability distributions of (0.0, 0.1, 0.1, 0.3, 0.2, 0.2, 0.1) and (0.1, 0.0, 0.4, 0.0, 0.3, 0.0, 0.2), respectively. Sources 1 has a maximum delay tolerance of 1 frame

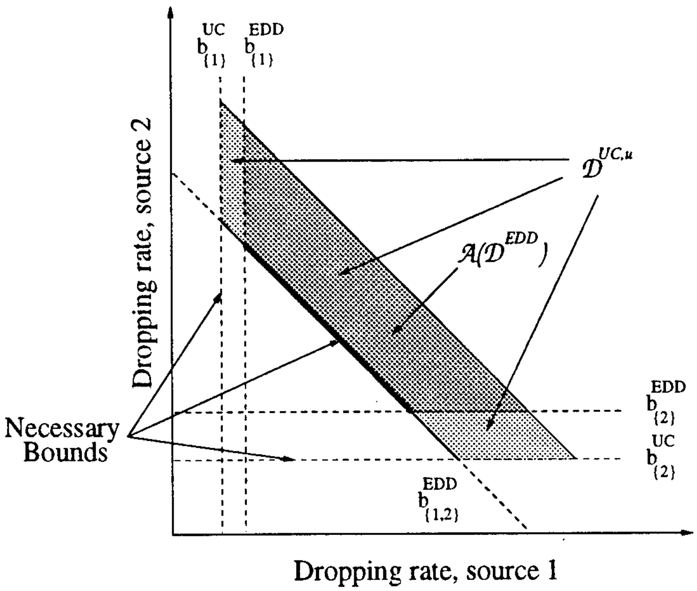


Figure 5: Acceptable QoS region under WC-EDD policies and its upper bound under any policy.

b_1^{EDD}	b_2^{EDD}	$b_{\{1.2\}}^{EDD}$
0.100	0.160	1.856

Table 1: Lower bounds on dropping rates under WC-EDD policies.

while sources 2 has a maximum delay tolerance of 2 frames.

The region of achievable QoS vectors under WC-EDD policies for this system is calculated from (11), (14), (18) and (21); the bounds on the dropping rates are given in Table 1. The unconstrained lower bounds were evaluated by applying (11) and (14) to each source k only, and the results are presented in Table 2. Notice that the unconstrained lower bound for source 1, b_1^{UC}, cannot be decreased beyond the WC-EDD lower bound, b_1^{EDD}, since every packet from source 1 has a maximum delay tolerance of 1 frame. The WC-EDD bound for source 2 can be decreased by relaxing the EDD condition on the system; that is, if the policy allows service to new arrivals from source 2 before packets from source 1, a lower dropping rate will be delivered to source 2. The decrease in the dropping rate beyond the WC-EDD bound for source 2 is achieved at the expense of increasing the system dropping rate. This result is verified through simulations and displayed in Fig. 6.

In the simulation results, policies[3] were generated that ranged from a policy that gives *all* packets from source 2 service priority over packets from source 1, to the DSO-HoL service policy π=(2,1) in which only the packets with a current service deadline from source 2 were given priority

[3] The performance of the DSO-HoL policy $\pi = (1,2)$ was also verified through simulation and is displayed in Fig. 6.

b_1^{UC}	b_2^{UC}
0.100	0.001

Table 2: Lower bounds on dropping rates under any policies.

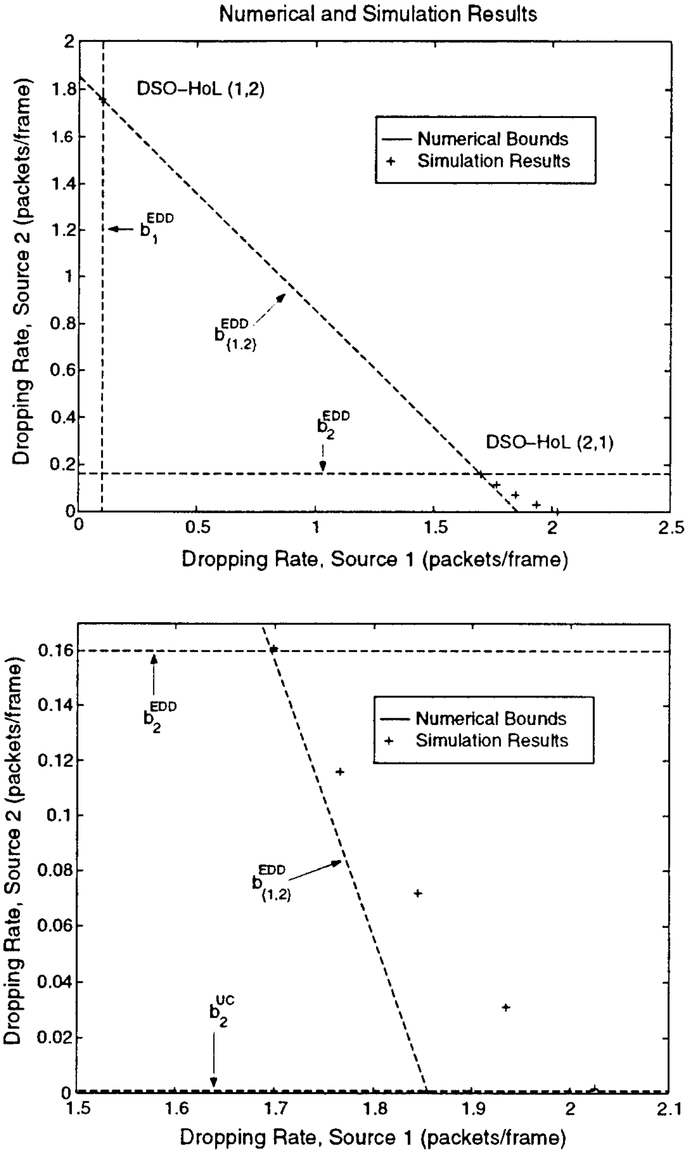


Figure 6: Evaluation of a numerical example with supporting simulation results.

State	Source 1	Source 2	Source 3
0	0.200	0.600	0.360
1	0	0	0
2	0.405	0.400	0.480
3	0	0	0
4	0.295	0	0.160
5	0	0	0
6	0.200	0	0

Table 3: State probability distribution for sources 1,2 and 3 of example 2.

over packet from source 1. Under the former policy, the UC lower bound ($b_2 = 0.001$) is achieved for source 2, but the system dropping rate is increased from its minimum of 1.859 to 2.025 packets per frame. The three policies which induce the other points shown in Fig. 6 were obtained by varying the frequency at which source 2 was allowed to violated the EDD condition. As is clearly observed in Fig. 6, for each of the four policies that violate the EDD condition, the resulting system dropping rate is increased compared to the conserved system dropping rate induced by WC-EDD policies.

As it was shown in section 3.2, satisfying the condition on the system performance, given by (27), is only necessary and not sufficient to guarantee that the target QoS vector is achievable[4]. To illustrate this concept, consider the system given in the previous example and let $\mathbf{d} = (0.099, 1.760)$ be the target QoS vector. $\mathbf{d}$ satisfies the condition on the system dropping rate, that is $d_{\{S\}} = b_{\{S\}}^{EDD} = 1.859$. Although, $d_1 < b_1^{UC} = 0.100$, and thus the target QoS vector cannot be achieved under any policy. For this system, the best possible performance source 1 can receive is if it is given absolute service priority, resulting in dropping rates of 0.100 and 1.759 for sources 1 and 2, respectively. The overall system performance is satisfied, but source 1 is experiencing poorer service than what is desired, while sources 2 is experiencing improved performance.

To illustrate the design of WC-EDD scheduling polices, consider the following example consisting of three VBR sources competing for 7 ($T = 7$) slots of a TDMA frame. In this example $M^1 = 3$, $M^2 = 5$ and $M^3 = 7$ and when in state i, a source generates i packets in the current frame. Each source is modeled by a sequence of i.i.d. random variables with a maximum arrival rate of 2, 4 and 6 packets per frame, respectively. The probability distributions for each source are given in Table 3.

Packets from source 1 have a maximum delay tolerance of 1 frame while packets from sources

[4] In the special case of a homogeneous system, such as a cellular voice system, satisfying (27) is sufficient to guarantee that the target QoS vector is achievable. This result has been established in [10].

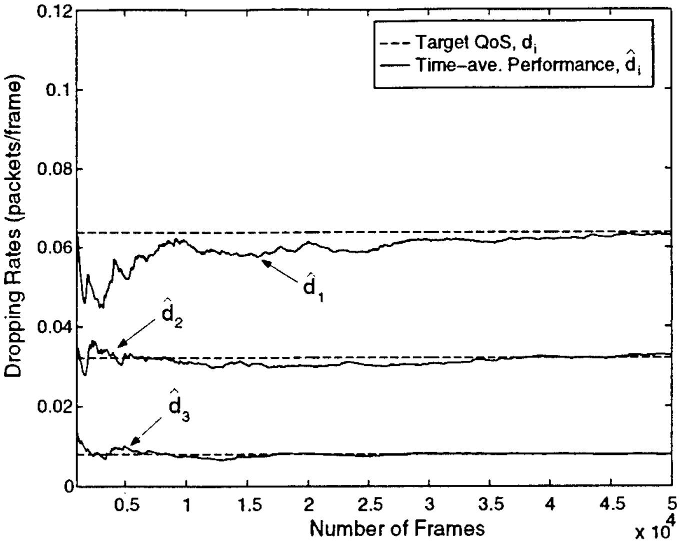


Figure 7: Time-averaged dropping rate for each source under the policy f_m.

2 and 3 can tolerate delays of up to 2 frames. The maximum dropping probabilities (resulting from delay violations) acceptable for sources 1, 2 and 3 are: $p_1 = 0.02$, $p_2 = 0.01$ and $p_3 = 0.02$. The QoS vector in this case is: $\mathbf{d}= (0.064, 0.008, 0.032)$.

By using (46), it can be determined that $\mathbf{d}$ is in the region of acceptable QoS vectors for WC-EDD polices. Furthermore, it can be shown that $\mathbf{d} \in \mathcal{D}^{EDD}$, and thus (as shown in section 5) there exists a mixing DSO-HoL policy $f_m \equiv \boldsymbol{\alpha}$, delivering it. Any $\boldsymbol{\alpha}$ satisfying (40)-(42) of Corollary 5.1 may be chosen. For this example, the following $\boldsymbol{\alpha}_o$ was chosen by employing linear programming techniques.

$$\boldsymbol{\alpha}_o = \begin{bmatrix} \alpha_1 & = 0 \\ \alpha_2 & = 0 \\ \alpha_3 & = 0.3367 \\ \alpha_4 & = 0.5386 \\ \alpha_5 & = 0 \\ \alpha_6 & = 0.1248 \end{bmatrix}. \qquad (47)$$

In Fig. 7, the time-averaged performance for each source under the selected mixing policy, f_m, is displayed. It can be seen that the target dropping rates, d_i, (and therefore dropping probabilities) are achieved for each source, $i = 1, 2, 3$.

As it is expected from the formulation of the constraints, more than one solution may be found. This allows for the incorporation of additional constraints representing other desirable qualities of the policies. Functions of interest may be minimized subject to the constraints presented in this paper to guarantee the achievability of the resulting policies. For instance, among all mixing policies inducing $\mathbf{d}$, the one which minimizes the variance in the service provided to certain sources may be identified. Such additional objectives will be pursued in the future.

9 Conclusion

In this work, the region of achievable QoS vectors was precisely determined for a system of heterogeneous VBR sources serviced under policies that are *work-conserving* and satisfy the *earliest due date* service criterion (WC-EDD). The QoS requirements for each application were defined in terms of a maximum tolerable packet delay and dropping probability. In addition to determining the region of achievable QoS vectors, a class of scheduling policies was developed which delivers any achievable performance in the region established for WC-EDD policies. The effectiveness of these policies in delivering the QoS vectors was verified through simulation. Also, an upper bound on the region of QoS vectors that can be achieved under any policy was determined. Numerical examples were presented with simulation results verifying the theoretical results presented in this paper.

10 Appendix

Lemma 10.1 *If λ_A (λ_B) are generic random variables representing the minimum residual traffic*[5] *from sources in set $A(B) \subseteq S$ in frame n (as defined in Lemma 3.1), then $\lambda_A(\lambda_B)$ satisfy the following.*

$$\begin{aligned} \lambda_A + \lambda_B &= \lambda_x(n) + \lambda^r_y(n) + \lambda_v(n) + \lambda^r_w(n) \\ &\leq \lambda_{\{x\cup v\}}(n) + \lambda^r_{\{y\cup w\}}(n) + \lambda_{\{x\cap v\}}(n) + \lambda^r_{\{y\cap w\}}(n) \\ &= \lambda_{\{A\cup B\}} + \lambda_{\{A\cap B\}}, \quad \forall A, B \subseteq S, \end{aligned} \tag{48}$$

where $x = \{A \cap S_1\}$, $y = \{A \cap S_2\}$, $\{v = B \cap S_1\}$ and $\{w = B \cap S_2\}$. □

Proof. Since $x, v \subseteq S_1$, $\lambda_x(n) + \lambda_v(n) = \lambda_{\{x\cup v\}}(n) + \lambda_{\{x\cap v\}}(n)$ and therefore (48) holds if,

$$\lambda^r_y(n) + \lambda^r_w(n) \leq \lambda^r_{\{y\cup w\}}(n) + \lambda^r_{\{y\cap w\}}(n) \quad \forall y, w \subseteq S_2. \tag{49}$$

As stated in Corollary 3.1, the number of service opportunities for the new arrivals, $\lambda_i(n)$, $i \in S_2$, (or excess resources) is independent of the policy $f \in \mathcal{F}$. Let X represent the number of such opportunities in any given frame. Therefore, by definition in (18), (49) is rewritten as,

$$\begin{aligned} &\max\{0, \lambda_y(n-1) - X\} + \max\{0, \lambda_w(n-1) - X\} \\ &\qquad \leq \max\{0, \lambda_{\{y\cup w\}}(n-1) - X\} + \max\{0, \lambda_{\{y\cap w\}}(n-1) - X\} \quad \forall y, w \subseteq S_2. \end{aligned} \tag{50}$$

As can be seen from (18), equality holds in (50) when $\lambda_{S_1}(n) + \lambda^r_{S_2}(n) > T$ since $X = 0$. When $\lambda_{S_1}(n) + \lambda^r_{S_2}(n) \leq T$ and,

[5] Residual traffic from sources in S_1 refers to the traffic that has a transmission deadline in the current frame. Thus, the minimum residual traffic for sources in set S_1 is just new arrivals from sources in S_1.

Case i: If $\lambda_{y\cap w}(n-1) \leq X$ and $\lambda_y(n-1)$ and/or $\lambda_w(n-1) \leq X$, then (50) holds[6].

Case ii: If $\lambda_{y\cap w}(n-1) > X \Longrightarrow \lambda_y(n-1), \lambda_w(n-1) > X$, then equality holds in (50).

Case iii: If $\lambda_{y\cap w}(n-1) \leq X$ and $\lambda_y(n-1), \lambda_w(n-1) > X$, then,

$$\begin{aligned} \lambda_y(n-1) - X + \lambda_w(n-1) - X &= \lambda_y(n-1) + \lambda_w(n-1) - 2X \qquad (51) \\ &= \lambda_{\{y\cup w\}}(n-1) - X + (\lambda_{\{y\cap w\}}(n-1) - X) \\ &\leq \lambda_{\{y\cup w\}}(n-1) - X + \max\{0, \lambda_{\{y\cap w\}}(n-1) - X\}, \end{aligned}$$

and thus proving Lemma 10.1. □

Lemma 10.2 *b_A (defined by (21)) is a super-modular set function; that is,*

$$b_A + b_B \leq b_{\{A\cup B\}} + b_{\{A\cap B\}} \qquad (52)$$

and equality holds only if and only if $A \subset B$ or $B \subset A$. □

Proof. Let λ_A (λ_B) be generic random variables representing the minimum residual traffic from sources in set $A(B) \subseteq S$ in frame n. Using Lemma 10.1, the following can be established.

$$\begin{aligned} &\left\{E\left[\lambda_A + \lambda_B \,\middle|\, \lambda_A + \lambda_B > T\right] - T\right\} P(\lambda_A + \lambda_B > T) \qquad (53) \\ &\leq \left\{E\left[\lambda_{\{A\cup B\}} + \lambda_{\{A\cap B\}} \,\middle|\, \lambda_{\{A\cup B\}} + \lambda_{\{A\cap B\}} > T\right] - T\right\} P\left(\lambda_{\{A\cup B\}} + \lambda_{\{A\cap B\}} > T\right), \\ &\qquad \forall A, B \subseteq S. \end{aligned}$$

Case (I): Assume A and B are disjoint ($A \cap B = \emptyset$), then (53) is written as,

$$\left\{E\left[\lambda_A + \lambda_B \,\middle|\, \lambda_A + \lambda_B > T\right] - T\right\} P(\lambda_A + \lambda_B > T) \leq b_{\{A\cup B\}}. \qquad (54)$$

The following implication is self-evident:

$$\lambda_A + \lambda_B > T \Longrightarrow \begin{cases} \lambda_A > T, \lambda_B \leq T \\ \lambda_A \leq T, \lambda_B > T \\ \lambda_A > T, \lambda_B > T \\ \lambda_A \leq T, \lambda_B \leq T, \lambda_A + \lambda_B > T. \end{cases}$$

Assuming that all sources are independent, then λ_A and λ_B are independent since sets A and B are disjoint. thus,

$$\begin{aligned} P(\lambda_A + \lambda_B > T) = P(\lambda_A > T)P(\lambda_B \leq T) + P(\lambda_A \leq T)P(\lambda_B > T) \qquad (55) \\ +P(\lambda_A \leq T, \lambda_B \leq T, \lambda_A + \lambda_B > T) \\ +P(\lambda_A > T)P(\lambda_B > T), \end{aligned}$$

[6] Under Case i, strict inequality holds if $\lambda_y(n-1)$, $\lambda_w(n-1) \leq X$ and $\lambda_{\{y\cup w\}}(n-1) > X$.

and therefore (54) may be written as,

$$\begin{aligned} b_{\{A\cup B\}} \geq & \left\{E\left[\lambda_A+\lambda_B \;\middle|\; \lambda_A > T, \lambda_B \leq T\right] - T\right\} P(\lambda_A > T) P(\lambda_B \leq T) \\ & + \left\{E\left[\lambda_A+\lambda_B \;\middle|\; \lambda_A \leq T, \lambda_B > T\right] - T\right\} P(\lambda_A \leq T) P(\lambda_B > T) \\ & + \left\{E\left[\lambda_A+\lambda_B \;\middle|\; \lambda_A > T, \lambda_B > T\right] - T\right\} P(\lambda_A > T) P(\lambda_B > T) \\ & + \left\{E\left[\lambda_A+\lambda_B \;\middle|\; \lambda_A \leq T, \lambda_B \leq T, \lambda_A+\lambda_B > T\right] - T\right\} \\ & P(\lambda_A \leq T, \lambda_B \leq T, \lambda_A+\lambda_B > T). \end{aligned} \tag{56}$$

Expanding the expected value operator over λ_A and λ_B and rearranging terms, (56) can be written as,

$$\begin{aligned} b_{\{A\cup B\}} \geq & \left\{E\left[\lambda_A \;\middle|\; \lambda_A > T\right] - T\right\} P(\lambda_A > T) + \left\{E\left[\lambda_B \;\middle|\; \lambda_B > T\right] - T\right\} P(\lambda_B > T) \\ & + E\left[\lambda_A \;\middle|\; \lambda_A \leq T\right] P(\lambda_A \leq T) P(\lambda_B > T) + E\left[\lambda_B \;\middle|\; \lambda_B \leq T\right] \\ & P(\lambda_A > T) P(\lambda_B \leq T) + T P(\lambda_A > T) P(\lambda_B > T) \\ & + \left\{E\left[\lambda_A+\lambda_B \;\middle|\; \lambda_A \leq T, \lambda_B \leq T, \lambda_A+\lambda_B > T\right] - T\right\} \\ & P(\lambda_A \leq T, \lambda_B \leq T, \lambda_A+\lambda_B > T) \\ & = b_{\{A\}} + b_{\{B\}} + \mathcal{K} \end{aligned} \tag{57}$$

where,

$$\begin{aligned} \mathcal{K} = & E\left[\lambda_A \;\middle|\; \lambda_A \leq T\right] P(\lambda_A \leq T) P(\lambda_B > T) + E\left[\lambda_B \;\middle|\; \lambda_B \leq T\right] \\ & P(\lambda_A > T) P(\lambda_B \leq T) + T P(\lambda_A > T) P(\lambda_B > T) \\ & + \left\{E\left[\lambda_A+\lambda_B \;\middle|\; \lambda_A \leq T, \lambda_B \leq T, \lambda_A+\lambda_B > T\right] - T\right\} \\ & P(\lambda_A \leq T, \lambda_B \leq T, \lambda_A+\lambda_B > T). \end{aligned} \tag{58}$$

Notice that $\mathcal{K} > 0$ (for the cases of interest corresponding to $T > 0$). Thus, from (57), and since $b_{\{A\cap B\}} = 0$ for $\{A \cap B\} = \emptyset$,

$$b_A + b_B < b_{\{A\cup B\}} + b_{\{A\cap B\}}. \tag{59}$$

or,

$$b_A + b_B = b_{\{A\cup B\}} + b_{\{A\cap B\}} - C^{A,B}, \tag{60}$$

where $C^{A,B} > 0$ is the additional expected number of dropped packets as a result of the competition for resources between sources in set A and B when $\{A \cap B\} = \emptyset$. The following property regarding $C^{A,B}$ is used to complete the proof.

Property 10.1

$$C^{\{A\cup x\},B} - C^{A,B} > 0, \tag{61}$$

when $\{A \cap B\} = \emptyset$, $x \notin \{A \cup B\}$ *and* $x \in S$. □

Proof. By definition of $C^{A,B}$ in (60) and by (21), (61) may be expressed as,

$$E\Big[\max(\lambda_{\{A \cup B \cup x\}} - T, 0) - \max(\lambda_{\{A \cup x\}} - T, 0) - \max(\lambda_B - T, 0) - \max(\lambda_{\{A \cup B\}} - T, 0) + \max(\lambda_A - T, 0) + \max(\lambda_B - T, 0)\Big] > 0. \tag{62}$$

Consider the following cases for each realization.

Case i: If $\lambda_A \geq T$ then by Lemma 10.1, (62) holds.

Case ii: If $\lambda_{\{A \cup x\}} \geq T$ and $\lambda_A \leq T$, then using Lemma 10.1,

$$\lambda_{\{A \cup B \cup x\}} - \lambda_{\{A \cup B\}} - \lambda_{\{A \cup x\}} + T \geq \lambda_x - \lambda_{\{A \cup x\}} + T \geq -\lambda_A + T > 0. \tag{63}$$

Case iii: If $\lambda_{\{A \cup B\}} \geq T$ and $\lambda_{\{A \cup x\}} < T$, then using Lemma 10.1,

$$\lambda_{\{A \cup B \cup x\}} - \lambda_{\{A \cup B\}} \geq 0. \tag{64}$$

Case iv: If $\lambda_{\{A \cup B \cup x\}} > T$ and $\lambda_{\{A \cup B\}} \leq T$, then

$$\lambda_{\{A \cup B \cup x\}} - T > 0. \tag{65}$$

Therefore, the random variable operated on by the expected value operator in (62) is non-negative. Provided any event that causes this random variable to have a positive value (as the ones indicated) has a non-zero probability (which is true for non-degenerate cases[7]), (62) is satisfied, thus proving Property 10.1. □

Case (II): Assume the A and B are not disjoint ($A \cap B \neq \emptyset$). By writing $\{A \cup B\}$ as the union of the disjoint sets $\{A\}$ and $\{B - A\}$ and using the results from Case I (60), the following can be written.

$$b_{\{A \cup B\}} = b_{\{A \cup \{B-A\}} = b_A + b_{\{B-A\}} + C^{\{A\},\{B-A\}} \tag{66}$$

Similarly, by writing set B as the union of the disjoint sets $\{A \cap B\}$ and $\{B - A\}$, the following can be written.

$$b_B = b_{\{A \cap B\}} + b_{\{B-A\}} + C^{\{A \cap B\},\{B-A\}}. \tag{67}$$

or,

$$b_{\{B-A\}} = b_B - b_{\{A \cap B\}} - C^{\{A \cap B\},\{B-A\}}. \tag{68}$$

Finally, by combining (66) and (68), the following is obtained,

$$b_{\{A \cup B\}} + b_{\{A \cap B\}} = b_A + b_B + C^{\{A\},\{B-A\}} - C^{\{A \cap B\},\{B-A\}}. \tag{69}$$

If $A \subset B$ then,

$$C^{\{A\},\{B-A\}} - C^{\{A \cap B\},\{B-A\}} = C^{\{A\},\{B-A\}} - C^{\{A\},\{B-A\}} = 0. \tag{70}$$

[7] A degenerate case is when there is no packet dropping in the system, that is $b_A = b_B = b_{\{A \cup B\}} = 0$.

Similarly if $B \subset A$. If $A \not\subset B$, then $\{A \cap B\} \subset A$ and by Property 10.1,

$$C^{\{A\},\{B-A\}} - C^{\{A\cap B\},\{B-A\}} > 0, \tag{71}$$

completing the proof of the lemma. □

Lemma 10.3 *The sets g_j, $j = 1, 2, \ldots, N$, that satisfy equation (29), defining the vertices of the polytope $\mathcal{D}^u$, must be strictly included in each other; that is*

$$\textit{either } h_j = g_j - (g_j \cap g_k) = \emptyset \textit{ or } h_k = g_k - (g_k \cap g_j) = \emptyset, \tag{72}$$

for all g_j and g_k in (29). □

Proof. Let g_1 and g_2 be two sets satisfying (29). By adding the corresponding equations the following is obtained.

$$b_{\{g_1\}} + b_{\{g_2\}} = \sum_{i \in g_1} d_i^* + \sum_{i \in g_2} d_i^* \;\; = \sum_{i \in \{g_1 \cup g_2\}} d_i^* + \sum_{i \in \{g_1 \cap g_2\}} d_i^* \tag{73}$$

or,

$$\sum_{i \in \{g_1 \cup g_2\}} d_i^* = b_{\{g_1\}} + b_{\{g_2\}} - \sum_{i \in \{g_1 \cap g_2\}} d_i^*. \tag{74}$$

Since for any subset of S (26) must be satisfied, then for $\{g_1 \cap g_2\}$, $\sum_{i \in \{g_1 \cap g_2\}} d_i^* \geq b_{\{g_1 \cap g_2\}}$, the following is obtained from (74),

$$\sum_{i \in \{g_1 \cup g_2\}} d_i^* \leq b_{\{g_1\}} + b_{\{g_2\}} - b_{\{g_1 \cap g_2\}}. \tag{75}$$

Suppose now that g_1 and g_2 are not strictly included in each other. That is $h_1 = g_1 - (g_1 \cap g_2)$ and $h_2 = g_2 - (g_1 \cap g_2)$ are both non-empty. Then, Lemma 10.2 implies that,

$$b_{\{g_1\}} + b_{\{g_2\}} < b_{\{g_1 \cup g_2\}} + b_{\{g_1 \cap g_2\}} \tag{76}$$

or,

$$b_{\{g_1\}} + b_{\{g_2\}} - b_{\{g_1 \cap g_2\}} < b_{\{g_1 \cup g_2\}}. \tag{77}$$

Therefore,

$$\sum_{i \in \{g_1 \cup g_2\}} d_i^* \leq b_{\{g_1\}} + b_{\{g_2\}} - b_{\{g_1 \cap g_2\}} < b_{\{g_1 \cup g_2\}} \tag{78}$$

or,

$$\sum_{i \in \{g_1 \cup g_2\}} d_i^* < b_{\{g_1 \cup g_2\}}, \tag{79}$$

which implies $\mathbf{d}^* \notin \mathcal{D}^u$, this is a contradiction since $\mathbf{d}^*$ is a vertex of $\mathcal{D}^u$. Thus the assumption that g_i are strictly included in each other follows. □

References

[1] D. Raychaudhuri and N. Wilson. ATM-Based Transport Architecture for Multiservices Wireless Personal Communication Networks. *IEEE Journal of Selected Areas Communications*, 12(8):1401–1414, October 1994.

[2] K.Y. Eng, M. Karol, and M. Veeraraghavan. A Wireless Broadband Ad-Hoc ATM Local Area Network. *ACM Wireless Networks Journal*, 1(2), December 1995.

[3] P. Mermelstien, A. Jalali, and H. Leib. Integrating Services on Wireless Multiple Access Networks. In *Proceedings of ICC*, pages 863–867, March 1993.

[4] D. Petras et al. MAC Protocol for Wireless ATM: Contention Free versus Contention Based Transmission reservation Requests. In *Proceedings of PIMRC*, Taipei, Taiwan, October 1996.

[5] M. Karol, Z. Liu, and K. Eng. An Efficient Demand-Assignment Multiple Access Protocol for Wireless Packet (ATM) Networks. *Wireless Networks*, 1(4):267–279, December 1995.

[6] A. Mahmoud, D. Falconer, and S. Mahmoud. A Multiple Access Scheme for Wireless Access to a Broadband ATM LAN Based on Polling and Sectored Antennas. *IEEE Journal on Selected Areas in Communications*, 14(4), May 1996.

[7] C. Chang, K. Chen, and M. You. Guaranteed Quality of Service Wireless Access to ATM Networks. In *ICC*, Dallas, TX, June 1996.

[8] D.J. Goodman, R.A. Valenzeula, et al. Packet Reservation Multiple Access for Local Wireless Communication. *IEEE Transactions on Communications*, 37(8):885–890, August 1989.

[9] D. Bantz and F. Baucho. Wireless LAN Design Alternatives. *IEEE Network Magazine*, 8(2):43–45, March-April 1994.

[10] J. Capone and I. Stavrakakis. Achievable QoS and Scheduling Policies in Integrated Services Wireless Networks. *Performance Evaluation*, 26 and 27(1), October 1996.

[11] S.S. Panwar, D. Towsley, and J.K. Wolf. Optimal Scheduling Policies for a Class of Queues with Customer Deadlines to the Beginning of Service. *Journal of ACM*, pages 832–844, 1988.

[12] G. Chen and I. Stavrakakis. ATM Traffic Management with Diversified Loss and Delay Requirements. In *Proceedings of IEEE INFOCOM*, San Francisco, CA, March 1996.

[13] Arne Brøndsted. *An Introduction to Convex Polytopes*. Springer-Verlag, 1983.

[14] E.G. Coffman and I. Mitrani. A Characterization of Waiting Time Performance Realizable by Single-Server Queues. *Operations Research*, 28(3):810–82, May-June 1980.

[15] W.W Sharkey. Cooperative Games with Large Cores. *International Journal of Game Theory*, 11:175–182, 1982.

[16] D.J.A. Welsh. *Matroid Theory*. Academic Press, 1976.

A Dynamic Regulation and Scheduling Scheme for Real-Time Traffic Management*

Steve Iatrou Ioannis Stavrakakis

Electrical & Computer Engineering
Northeastern University
Boston, MA 02115
USA

Abstract

Typical rate-based traffic management schemes for real-time applications attempt to allocate resources by controlling the packet delivery to the resource arbitrator (scheduler). This control is typically based only on the characteristics of the particular (tagged) traffic stream and would fail to optimally adjust to non-nominal network conditions such as overload. In this paper, a dynamic Regulation and Scheduling (dynamic-R&S) scheme is proposed whose regulation function is modulated by both the tagged stream's characteristics and some information capturing the state of the co-existing applications as provided by the scheduler. The performance of the proposed scheme – as well as the equivalent static one – is investigated under both underload and overload traffic conditions and the substantially better throughput / jitter characteristics of the dynamic-R&S scheme are established.

Please send correspondence to:

Prof. Ioannis Stavrakakis
ECE Dept., 409 Dana Building, Northeastern University, Boston, MA 02115, USA
Email: ioannis@cdsp.neu.edu

*Research supported in part by the Defense Advanced Research Project Agency under Grant F49620-93-1-0564 monitored by the Air Force Office of Scientific Research and the National Science Foundation under Grant NCR-9628116.

1 Introduction

A secure solution to the problem of guaranteeing the QoS of real-time applications will typically require the reservation of the maximum amount of needed resources. Because of the anticipated low - due to resource requirement fluctuation - network utilization, alternative solutions are being considered based on "over allocation" of resources to a group of applications (multiplexing). Over allocation aims at improving the resource utilization, at the expense of failing to accommodate at certain times peak resource demands from a number of the grouped applications. Or, each application cannot expect to always be provided resources to accommodate its current demand. Grouping of applications allows for allocation of the resources to the participating applications without reserving them by individual applications and potentially wasting them if not used by this application. Grouping of applications and "over-allocation" of resources are the key aspects of non-degenerate statistical multiplexing.

A basic - and widely considered - statistical multiplexing scheme is the one which buffers incoming packets (or cells) in a First-Come, First-Stored fashion and schedules them in a First-Come, First-Served fashion. This basic scheme - or simple variations to it - have been adopted and studied extensively in traditional data networking environments. Because of the stringent QoS requirements of real-time applications, it is expected that such "traditional" statistical multiplexing schemes will not be effective for such applications. It is well understood that some tighter control should be exercised on input, output as well as in the internal processes of a multiplexing scheme which impact on its efficiency. Through (a) sophisticated call admission schemes or other types of "weak" resource reservation at the larger time scale and (b) proper traffic regulation and service scheduling mechanisms at the smaller time-scale, statistical multiplexing will potentially provide for increased network utilization while delivering the more stringent QoS associated with real-time applications. This paper is focused on the control approaches at the smaller time scale, namely regulation and scheduling.

Typical QoS metrics shaped by regulation and scheduling schemes are described in terms of the induced cell loss/delay and the allocated bandwidth. In principle, a target value of such QoS metric may be possible to achieve through either tight traffic regulation (typically referred to as rate-based approach), or sophisticated scheduling (typically referred to as scheduler-based approach) only. In most practical cases though, some scheduling will be needed to resolve transmission conflicts among rate-based controlled applications. Similarly, some traffic filtering (regulation) will be needed to eliminate extreme traffic realization which would be hard to manage even by a sophisticated scheduler under a scheduler-based approach. In addition to the supporting role of regulation in a scheduler-based approach, and scheduling in a rate-based approach, higher multiplexing gain may be achieved by allowing for some cooperation between the two functionalities which will typically co-exist and be designed to achieve a common goal. Such a dynamic policy is investigated in the present paper.

Substantial effort has been directed toward the development of regulation and scheduling schemes for real-time applications. Rate-based schemes attempt to provide for the target QoS metric by controlling the bandwidth demands of the supported applications through rate regulation; some simple (static) priority FCFS scheduling may be employed for transmission conflict resolution. Scheduler-based schemes typically associate deadlines with cells and attempt to provide for the target QoS metric by scheduling service in order of increasing deadlines. Examples of regulation and scheduling schemes for real-time applications include : D-EDD (Delay-Earliest Due Date) [1]; J-EDD (Jitter Earliest Due Date) [2], [3]; HRR (Hierarchical Round Robin) [4]; S&G (Stop and Go Queuing) [5]; WFQ (Weighted Fair Queuing) [6]; PGPS (Packet Generalized Processor Sharing) [7]; RCSP (Rate Controlled Static Priority) [8]; LIT (Leave In Time) [9]; MRTS (Multi Rate Traffic Shaping) [10]; VC (Virtual Clock) [11].

The traffic management scheme for real-time applications investigated in this work may be viewed as an enhancement of the Rate Controlled Static Priority (RCSP) scheme proposed in [8]. Under the RCSP scheme each traffic stream passes through a regulator which restores the traffic, completely or partially, based on the traffic description and the type of regulator used. The restored traffic is handed over to the respective priority queue and is scheduled in FCFS order. The rate jitter regulator employed in [8] is based on cell eligibility times (ET) defined as follows: $ET_1 = AT_1$; $ET_k = max\{ET_{k-1} + T + \tau_k, AT_k\}$, $k > 1$ where AT denotes the cell arrival time, and τ_k is a term used to provide the average rate; subscripts indicate packets. Rate jitter is controlled with respect to the ET of the previous packet of the same connection. T is the minimum cell inter arrival time specified by the source. The idea is to hold cells so that minimum inter departure time be enforced.

An apparent drawback of schemes such as the RCSP is that the regulator and scheduling functions are separated. It is expected though that the throughput / jitter of a regulated tagged stream will potentially be substantially modulated at the scheduler by the cumulative activity of the co-existing traffic streams. As a consequence the effectiveness of the Regulation and Scheduling (R&S) scheme may be compromised significantly. This problem can be addressed to some extent by *dynamically* adjusting the regulator behavior based on state information fed back from the scheduler to the regulator.

In section 2 the proposed dynamic R&S (dynamic-R&S) policy is motivated and presented, along with the equivalent static R&S (static-R&S) policy on which a comparative study will be based. In section 3 the behavior of the tagged regulator is investigated under both policies. In section 4, the behavior of the scheduler is studied under underload conditions at the scheduler. The tagged cell interdeparture process at the scheduler is derived by formulating and analyzing an interesting queueing system in which arrivals depend on the queue occupancy. Based on this process, the throughput / jitter performance of the policies is evaluated. In section 5 the throughput / jitter performance of the policies is investigated under overload conditions at the scheduler. Numerical results illustrating the performance of the dynamic-R&S and static-R&S policies under underload and overload conditions at the scheduler are presented in section 6. These results are based on the analytical studies presented earlier in the preceding sections and are in accordance with expectations. Finally, some of the assumptions made in order to facilitate the analysis are relaxed in section 7 where simulation results are presented involving multiple sources subject to the R&S policies considered in this paper.

2 The Proposed Dynamic R&S Policy

The typical primary objective in regulating real-time traffic stream within the network is to control jitter or the instantaneous rate (throughput). This is achieved in the RCSP mechanism [8] by enforcing a minimum spacing at the output of the regulator associated with the traffic stream of interest (tagged traffic stream). Fig. 1 (without the feedback loop) shows a block diagram of an architecture implementing a RCSP mechanism where each of the N multiplexed streams is regulated before it is considered for transmission.

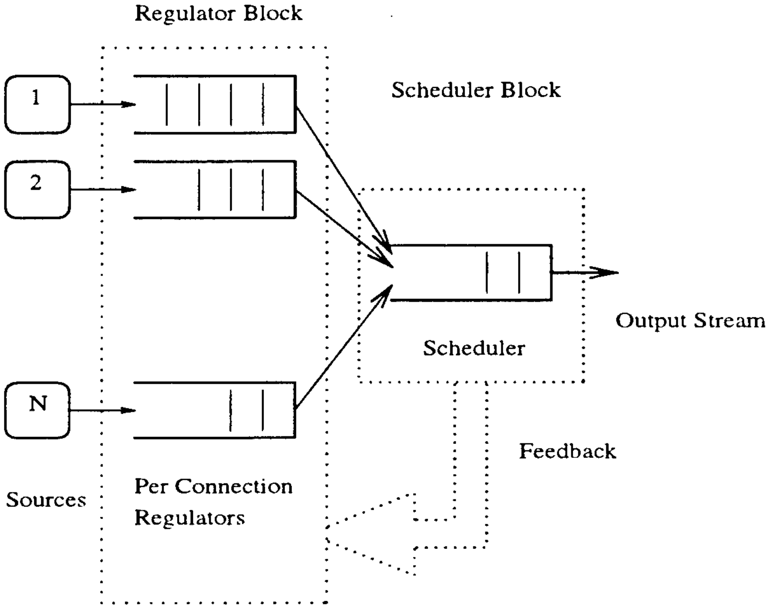


Figure 1: The dynamic-R&S (including the arrow) and the static-R&S (excluding the arrow) Systems

Since scheduling conflicts will arise when more than one regulated applications are present, a scheduler needs to be employed to resolve these conflicts. A consequence of the scheduling conflicts is that the tagged traffic stream at the output of the scheduler will be a distorted version of the target stream enforced at the output of the regulator. For instance, although a minimum spacing between consecutive tagged cells is enforced at the output of the tagged regulator in Fig. 1, this does not hold true for the tagged stream at the output of the scheduler. This clustering is generated due to an increased arrival rate to the scheduler in the immediate past which has pushed back (delayed) earlier tagged cells. Due to the latter, some spreading followed by some clustering of tagged cells is expected to be observed at the output of the scheduler.

The tagged cell spreading mentioned above can be reduced by monitoring the scheduler and releasing a tagged cell before its eligibility time [1] when scheduler queue build ups, which will cause the spreading, are detected. The dynamic Regulation and Scheduling (dynamic-R&S) scheme proposed below attempts to provide for a smoother tagged traffic at the output of the scheduler based on this

[1] Here defined as T time units following the previous tagged cell release, if a minimum spacing of T is targeted.

idea.

Although less commonly stated, another objective in regulating real-time traffic streams within the network is to control (limit) the amount of bandwidth that is demanded by traffic streams. This is a mechanism to protect other real-time applications from excessive bandwidth consumption by "misbehaving" ones. This "misbehavior" is typically due to the traffic fluctuations (burstiness). Although severe traffic fluctuations will be eliminated by the regulator(s), some traffic burstiness will remain since the regulator(s) will not typically attempt to transform Variable Bit Rate (VBR) to Continuous Bit Rate (CBR) traffic, due to the delay constraints. For instance, remaining burstiness is present in a traffic stream regulated by a leaky bucket, or when high multiplexing gain is to be achieved. Such controlled burstiness will potentially be "absorbed" in part by the multiplexing process.

When the arrival rate to the scheduler increases due to (residual) traffic burstiness, spreading of the tagged cells at the output of the scheduler is expected to be observed, as indicated again above. This spreading represents an instantaneous reduction in the bandwidth allocated to the tagged traffic stream, as measured at the output of the scheduler. In the context of the bandwidth availability to the tagged traffic stream, the dynamic-R&S scheme proposed below may be viewed as attempting to provide for a constant bandwidth availability to the tagged traffic stream at periods of excessive total bandwidth demand from the co-existing applications.

A simple architecture of the switch for the illustration of the proposed policy is shown in Fig. 1. Similarly to the architecture proposed in [8], each of the N supported real-time applications is regulated at a logically dedicated regulator before it is delivered to the scheduler. In the present work, a simple FCFS scheduling policy is being considered. This scheduling mechanism is the simplest possible, reducing the scheduling complexity to single queue buffering.

Under the dynamic-R&S policy proposed below, the regulation process is modulated by some scheduler status information. Unlike in past work in the area, appropriate information regarding the status of the scheduler (FCFS queue) is fed back to the regulators, as indicated in Fig. 1 with the feedback arrow. As explained below the time of delivery of the cell from regulator i to the scheduler initiates a cycle of scheduler status monitoring for regulator i; this cycle is completed with the delivery of the next cell from regulator i to the scheduler, initiating the next monitoring cycle. The time at which a tagged cell is shifted from the regulator to the scheduler is determined by the tagged cell release policy described below.

2.1 The tagged cell release policy: dynamic-R&S scheme

Let t_k denote the time slot at which the k^{th} tagged cell is released from the tagged regulator. Let Q_k^r denote the queue occupancy at the regulator upon (following) the release of the k^{th} cell. Let $t_k + B_k$ ($B_k \geq 1$) denote the time slot at which the cumulative number of non-tagged arrivals (releases) to the scheduler following t_k exceeds $T - 2$ for the first time. Let a superscript d (s) indicate a quantity associated with the dynamic-R&S (static-R&S) policy and let

$$W_k = min\{B_k, T\} \quad \text{or} \quad W = min\{B, T\}, \tag{1}$$

where the last expression involves the generic random variables W and B. The $(k+1)$st tagged cell release time t_{k+1} is given by

$$t_{k+1} = t_k + W_k + H_k^d * 1_{\{Q_k^r=0, A_r^{d,W_k}=0\}}, \tag{2}$$

where H_k^d denotes the time interval between $\overline{t_k} = t_k + W_k$ and the first tagged cell arrival following $\overline{t_k}$; T is a constant positive integer; $A_r^{d,j}$ is the number of cell arrivals to the dynamic regulator over j slots.

If T is equal to the minimum spacing among consecutive tagged cell releases from the regulator in the RCSP scheme [8], then it is easy to see that the above release policy will accelerate the tagged cell releases from the regulator at times when a minimum spacing of T at the output of the scheduler would be violated. This acceleration occurs when $B_k < T$.

It is expected that the tagged cell release acceleration will have a positive impact on the tagged cell delay jitter and availed bandwidth. To quantify such benefits the static-R&S scheme is considered in parallel in the rest of the paper. As described below its tagged cell release policy is not modulated by any scheduler status information. A simple FCFS scheduler is also considered.

2.2 The tagged cell release policy: static-R&S scheme

By employing the definitions presented above and replacing $W_k = min\{B_k, T\}$ by T, the $(k+1)$st tagged cell release time t_{k+1} is given by:

$$t_{k+1} = t_k + T + H_k^s * 1_{\{Q_k^r=0, A_r^{s,T}=0\}} \tag{3}$$

where H_k^s denotes the time interval between $\overline{t_k} = t_k + T$ and the first tagged cell arrival following $\overline{t_k}$; $A_r^{s,j}$ is the number of cell arrivals to the static regulator over j slots.

3 Study of Regulator Behavior

The behavior of the R&S schemes is evaluated by investigating their impact on a specific stream (tagged stream). The traffic at the output of the regulators associated with the remaining $N-1$ applications is aggregated and forms the background traffic which competes with the tagged traffic for resources at the scheduler. Let A^k denote the number of background cells delivered to the scheduler over k consecutive slots; let A_i denote the number of background cells delivered to the scheduler in the ith slot (Note that $A^k = \sum_{i=1}^{k} A_i$).

The improved performance of the dynamic-R&S policy over the static-R&S policy will be established by evaluating the cell interdeparture process at the output of the scheduler. Since the input process to the scheduler is the output process from the regulator, it is important that the latter be investigated to both gain insight into the combined system (regulator plus scheduler) behavior as well as evaluate the output process at the scheduler.

The basic operational difference between the dynamic-R&S and static-R&S schemes is captured by the tagged cell interdeparture process from the regulator $\{V_k\}_{k\geq1}$, where $V_k = t_{k+1} - t_k$. In view of (2) and (3) it is easy to establish that the evolution of the tagged cell process $\{V_k\}_{k\geq1}$ is described by

$$V_k^s = T + H_k^s * 1_{\{Q_k^r=0, A_r^{s,T}=0\}} \qquad \text{(static-R\&S scheme)} \tag{4}$$

$$V_k^d = W_k + H_k^d * 1_{\{Q_k^r=0, A_r^{d,W_k}=0\}} \qquad \text{(dynamic-R\&S scheme)} \tag{5}$$

In order to decouple the intrinsic behavior – to be investigated in this paper – of the R&S schemes from the source load, the heavy traffic source assumption will be made throughout the paper. This assumption is consistent with standard ones made in order to determine the throughput capabilities of a scheme as well as the throughput fluctuations (jitter), without the noise introduced by source inactivity periods. Under the heavy traffic source assumption the indicator function in (4) and (5) is always zero and, thus,

$$V_k^s = T \qquad \text{(static-R\&S)} \tag{6}$$

$$V_k^d = W_k \qquad \text{(dynamic-R\&S)} \tag{7}$$

The following proposition describes the generic random variable W.

Proposition 1 : *The probability mass function of W and its mean are given by:*

$$Pr\{W = j\} = \tilde{F}_{j-1}(T-2) - \tilde{F}_j(T-2) \ , \quad 1 \leq j \leq T \tag{8}$$

$$E\{W\} = \sum_{j=1}^{T} \tilde{F}_{j-1}(T-2) \tag{9}$$

where $\tilde{F}_0(T-2) \overset{\text{def}}{=} 1$, $\tilde{F}_T(T-2) \overset{\text{def}}{=} 0$ and $\tilde{F}_j(T-2) \overset{\text{def}}{=} F_j(T-2)$, $1 \leq j < T$, where $F_j(\cdot)$ denotes the j-fold convolution of the probability distribution function of A^1.

The proof of this proposition along with the proofs of other propositions and corollaries that follow may be found in the Appendix.

Proposition 1 describes the regulator interdeparture process of the dynamic-R&S scheme in terms of the first order probability mass function and the first moment. This process will be employed in the study of the scheduler under the dynamic-R&S scheme. Its description is also employed in the following comparative study of the two policies.

Proposition 2 : *The maximum throughput (output) rate of the tagged regulator under the two policies is given by*

$$R^s_{max} = \frac{1}{T} \tag{10}$$

$$R^d_{max} = \frac{1}{E\{W\}} = \frac{1}{\sum_{j=1}^{T} \tilde{F}_{j-1}(T-2)} \tag{11}$$

Notice that $R^d_{max} \geq R^s_{max}$ with equality only when the background traffic process can never deliver more than $T-2$ cells over $T-1$ consecutive slots (typically, a zero probability event).

The above discussion establishes that the dynamic-R&S scheme will respond to a sudden increase of the background load [2] by increasing its rate above the targeted rate of $1/T$, in an effort to ensure that the targeted tagged rate at the output of the scheduler is achieved. The impact of such a reaction (which is not possible under the static-R&S scheme) on the scheduler output process is investigated in the next section.

4 Study of Scheduler Behavior under Underload Conditions

In this section, the tagged cell interdeparture process from the scheduler is derived in order to evaluate the throughput / jitter properties of the R&S schemes. Since the deliverability of the QoS of the tagged application (which is considered to be a measure of the induced throughput / jitter) is expected to be decreased under high load conditions at the scheduler, the latter will be studied under high load conditions both below (underload) and above (overload) the scheduler capacity. Although the call admission control function will attempt to minimize the occurrence of temporary overload at the network nodes, due to traffic burstiness and the desire to achieve high resource utilization through statistical multiplexing, it is expected that temporary overload will be unavoidable. It is under such conditions that traffic management mechanisms should not only not collapse, but rather minimize the impact of the overload on the QoS. In order to evaluate the tagged output process at the scheduler as shaped by the R&S policy as opposed to buffer overflows, sufficiently large buffer will be assumed to be available at the scheduler throughout this paper, to eliminate the interference from the buffer overflow process.

Under underload conditions ($\rho < 1$) the scheduler queue is stable and since no cell overflow is possible the following relationships between the maximum regulator (R_{max}) and scheduler (S_{max}) throughputs hold.

$$R^d_{max} = S^d_{max} \quad \text{and} \quad R^s_{max} = S^s_{max} \tag{12}$$

In view of the above and Proposition 2 the following corollary is self evident.

[2] which would result in an instantaneous reduction of the tagged throughput at the output of the scheduler

Corollary 1 : *$S^d_{max} \geq S^s_{max}$, where S_{max} denotes the maximum tagged cell output rate from the scheduler, or the maximum tagged cell throughput.*

Corollary 1 implies that potentially higher throughput will be achieved by the tagged application under the dynamic-R&S policy compared to that under the static-R&S policy. This increase in throughput is due to a regulator tagged interdeparture interval W less than T under the dynamic-R&S policy, occurring only if the tagged cell interdeparture from the scheduler is to exceed the tagged value of T. Since the R&S policy attempts to minimize the variation of the target spacing T between consecutive tagged cell departures (or minimize the deviation of the instantaneous tagged cell output rate from $1/T$), the deviation of the tagged cell interdeparture interval from the target T will be employed as a jitter metric. The precise study of the tagged interdeparture process at the output of the scheduler – denoted by $\{X_k\}_{k\geq 1}$ – is presented in the next subsections under both policies.

To avoid unnecessary complications as well as evaluate the performance of the policies under non-idling environment - in which case they differ - the heavy traffic source assumption is employed here for the tagged traffic at the input of the regulator, as indicated earlier. That is, the tagged cell release to the scheduler process will be dependent only on the policy and the scheduler state and not on the regulator queue occupancy; the latter is considered to be non-empty under the heavy traffic assumption.

4.1 Tagged cell interdeparture under the dynamic-R&S policy

A queueing model for the system of the tagged traffic regulator and scheduler is shown in Fig. 2; the regulator (C^R) and scheduler (C^S) buffer capacities are assumed to be theoretically infinite as mentioned earlier. As indicated earlier, the heavy traffic assumption implies the regulator will not starve and thus V_k depends only on $W_k = min\{B_k, T\}$, where B_k is the shortest time interval (in slots), initiated upon the release of the k^{th} tagged cell and terminated at the slot in which the cumulative non tagged (or background) arrivals over this interval exceed $T - 2$.

As implied in Fig. 2, $N - 1$ traffic streams - potentially controlled by a similar R&S scheme - are assumed to compete for the common, slotted transmission resource. For analysis tractability, the cumulative arrivals from the $N - 1$ co-existing traffic streams - forming the background traffic - are assumed to be independent (per slot) and identically distributed, represented by the random variable A^1 (see earlier); the maximum number of background arrivals per slot is equal to N-1.

Fig. 3 illustrates the sequence of events associated with a slot, as considered in this study, and presents other time-related conventions. A cell departure (if any) is assumed to occur first, followed by the cumulative background arrivals over this slot (if any) and then the tagged cell arrival (if any). $X_k - 1$ slots are available to the background traffic between two consecutive tagged cell departures with interdeparture interval equal to X_k.

The scheduler interdeparture time between tagged cells k and $k + 1$, X_k, can be determined based on the scheduler queue occupancy, Q_k, found upon arrival of the tagged cell k to the scheduler queue

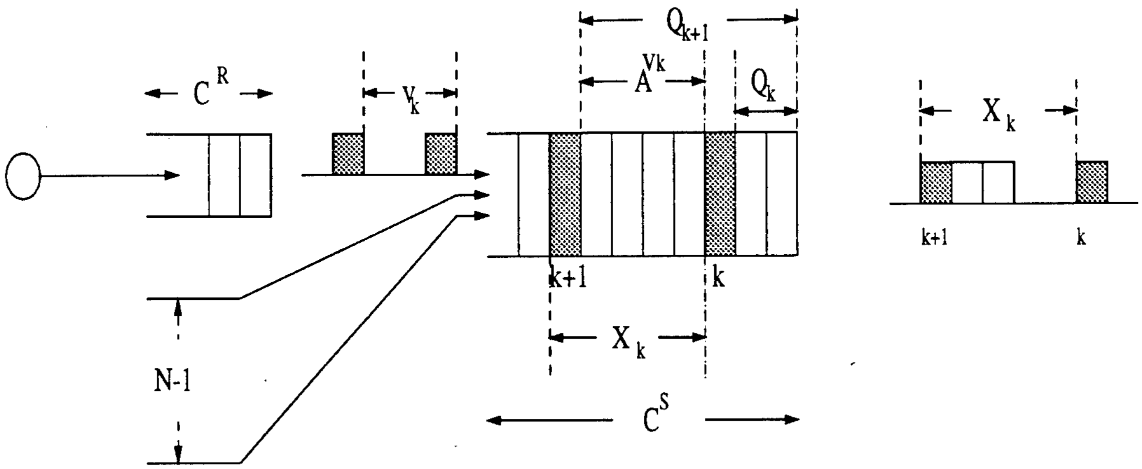


Figure 2: Queuing model for the tagged traffic study at the scheduler

(not counting itself). This is described next. Later, the stationary probabilities of Q_k are derived to complete the calculation of the probability distribution of X_k.

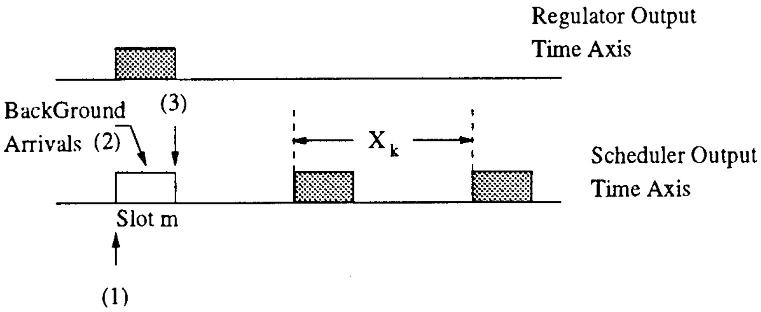


Figure 3: Sequence of events in the scheduler output slot m : (1) cell departure; (2) background cell arrivals; (3) tagged cell arrival

A rarely encountered peculiarity of the scheduler queue is that tagged cell arrivals depend on the queue build up, or, the queue's arrival process depends on its occupancy process. Specifically, tagged cell arrival $k+1$ occurs T slots following the arrival of tagged cell k if and only if the cumulative background arrivals following ("behind") the tagged cell k arrival have not exceeded $T-1$ at any earlier slot. Otherwise, tagged cell $k+1$ will arrive (be released from the regulator) at that earlier slot. The tagged cell k will be served in (during) the (Q_k+1)st slot following its release from the regulator. The following proposition provides for the conditional value of X_k given Q_k.

Proposition 3 *The conditional probability of X_k given Q_k, $Pr\{X_k = T + l/Q_k = i\}$ for $-(T-1) \leq l \leq (N-2)$, is given by the following expressions:*
Case I : $l \geq 0$ (no clustering)

$$Pr\{X_k = T + l/Q_k = i\} \quad = \quad \sum_{m=1}^{min\{i+1,T\}} Pr\{A^m = T + l - 1, A^{m-1} \leq T - 2\} +$$

$$+1_{\{i\le T-2\}} * \sum_{m=0}^{T-2} Pr\{A^{i+1}=m, X_k = T+l\} \tag{13}$$

Case II : $l<0$ (clustering)

$$\begin{aligned} Pr\{X_k = T+l/Q_k = i\} &= 1_{\{i\ge T-1\}} * Pr\{A^T = T+l-1\} + \\ &+1_{\{i\le T-2\}} * \sum_{m=0}^{T-2} Pr\{A^{i+1}=m, X_k = T+l\} \end{aligned} \tag{14}$$

where the probabilities involved in the above expressions are derived and evaluated in the proof of this proposition. □

The transition probabilities of the scheduler occupancy process $\{Q_k\}_{k\ge 1}$ are given in the following proposition.

Proposition 4 *The transition probabilities of the Markov Process $\{Q_k\}$, embedded upon tagged cell arrival times, are given by the following expressions :*
Case I : $i \ge T-1$

$$\begin{aligned} &Pr\{Q_{k+1} = j/Q_k = i\} = \\ &\quad = Pr\{A^1 = j-i\} * 1_{\{T-1\le j-i\le N-1\}\cap\{T-1\le N-1\}} + \\ &\quad + \sum_{m=2}^{T-1} Pr\{A^m = j-i-1+m, A^{m-1}\le T-2\} * 1_{\{T-m\le j-i\le T-2+N-m\}\cap\{T-1\le m*(N-1))\}} + \\ &\quad + Pr\{A^T = j-i-1+T\} * 1_{\{-(T-1)\le j-i\le N-2\}} \end{aligned}$$

Case II : $i\le T-2$

$$\begin{aligned} &Pr\{Q_{k+1} = j/Q_k = i\} = \\ &\quad Pr\{A^1 = j-i\} * 1_{\{T-1\le j-i\le N-1\}\cap\{T-1\le N-1\}} + \\ &\quad + \sum_{m=2}^{i+1} Pr\{A^m = j-i-1+m, A^{m-1}\le T-2\} * 1_{\{T-m\le j-i\le T-2+N-m\}\cap\{T-1\le m*(N-1))\}} + \\ &\quad + \sum_{m=0}^{T-2}\sum_{n=1}^{T-i-2}\sum_{a_n=T-1-m}^{T-3+N-m} Pr^n\{\hat{A}(n) = a_n, \hat{Q}(n) = j/\hat{A}(0)=0, \hat{Q}(0)=m\} \\ &\quad * Pr\{A^{i+1} = m, A^i \le T-2\} + \\ &\quad + \sum_{m=0}^{T-2}\sum_{a=0}^{T-3+N-m} Pr^{T-i-1}\{\hat{A}(T-i-1) = a, \hat{Q}(T-i-1) = j/\hat{A}(0)=0, \hat{Q}(0)=m\} * \\ &\quad * Pr\{A^{i+1} = m, A^i \le T-2\}\square \end{aligned} \tag{15}$$

Finally the tagged cell interdeparture probability distribution can be obtained from the conditional ones (Proposition 3) and the stationary probabilities of $\{Q_k\}$, $\pi_q(i)$, derived by employing the transition probabilities given in Proposition 4. Thus,

$$Pr\{X_k = T + l\} = \sum_{i=0}^{\infty} Pr\{X_k = T + l/Q_k = i\} * \pi_q(i)$$
$$= \sum_{i=0}^{T-2} Pr\{X_k = T + l/Q_k = i\} * \pi_q(i) +$$
$$+[1 - \sum_{i=0}^{T-2} \pi_q(i)] * Pr\{X_k = T + l/Q_k \geq T - 1\} \tag{16}$$

Since the conditional interdeparture given $Q_k = i$ for $i \geq T-1$ is constant, independent from i (see Proposition 3), equation (16) suggests that since only the stationary probabilities $\pi_q(i)$ for $i \leq T-2$ are used, truncating $\{Q_k\}_{k\geq 1}$ to some state $T-2+L$, for some large L, will cause negligible impact on the $\pi_q(i)$ for $i \leq T-2$ since only values away from the truncation boundary are used in (16).

4.2 Tagged cell interdeparture under the static-R&S policy

Under the static-R&S policy, no early tagged cell releases are allowed, and therefore the scheduler is fed by a periodic tagged traffic of period T (heavy traffic assumption at the regulator).

Proposition 5 : *The probability distribution of the jitter for the* static-R&S *policy is given by:*

$$Pr\{X_k = T + l\} = \sum_{j=1}^{\infty} Pr\{Q_{k+1} = i + l/Q_k = i\} * \pi_q(i) \tag{17}$$

The transition probabilities $Pr\{Q_{k+1} = j/Q_k = i\}$ can be obtained as the T-step transition probabilities of a simple $M/D/1$ queue. Finally by employing the transition probabilities, the stationary distribution $\pi_q(i)$ can be obtained.

5 Study of Scheduler Behavior under Overload Conditions

Since the traffic streams which are traversing a single multiplexer can be of highly variable rates, it is possible - for a short period of time - to have an average arrival higher than 1. If this condition were to persist, then the buffers would grow without bound and the system would be unstable. It is desirable to study the behavior of the policies under such extreme conditions. The case in which the scheduler queue is unstable is considered next. That is, $\rho > 1$.

Although the scheduler queue capacity is again assumed to be infinite and, for that matter, no overflow will occur, the relationship in (12) will not hold since traffic will accumulate in the infinite queue. This ever increasing queue built up will represent a difference between input and output traffic rate, making it hard to assess the precise throughput achieved by the tagged or background streams.

In view of the large buffer assumption at the scheduler it is evident that the scheduler queue occupancy upon tagged cell arrival, Q_k, will always exceed $T-1$ under overload conditions ($\rho > 1$). As a consequence – as it will become evident below – the analysis of the scheduler tagged cell interdeparture process is simplified significantly. For this reason both policies are treated concurrently.

The following proposition provides for the precise description of the tagged stream interdeparture distribution at the output of the scheduler under overload conditions.

Proposition 6 : *Under the overload conditions at the scheduler queue ($\rho > 1$) and infinite scheduler buffer capacity, the distribution of the tagged cell interdeparture at the scheduler, X_k, under the* dynamic-R&S *policy, is given by:*

$$Pr\{X_k = l\} = \sum_{m=1}^{T} Pr\{A^m = l-1, A^{m-1} \leq T-2\} * 1_{\{T \leq l \leq N+T-3\}} + \\ +Pr\{A^T = l-1\} * 1_{\{1 \leq l \leq T-1\}} \quad \textit{for } 1 \leq l \leq N+T-3; \tag{18}$$

the distribution under the static-R&S *policy is given by :*

$$Pr\{X_k = l\} = Pr\{A^T = l-1\} \quad \textit{for } 1 \leq l \leq (N-1)*T+1 \tag{19}$$

Proposition 6 can be employed in deriving the throughput of the tagged application. The following proposition establishes the better jitter and smoothness characteristics of the dynamic-R&S scheme, compared to those of the static-R&S scheme under overload conditions at the scheduler.

Proposition 7 : *Under overload conditions at the scheduler the* dynamic-R&S *policy will potentially reduce the tagged cell spreading (while it will never increase it) compared to the* static-R&S *policy. That is,*

$$Pr\{X_k^d > m\} \leq Pr\{X_k^s > m\} \qquad \textit{for } m > T.$$

The tagged cell clustering will be identical under both policies, that is,

$$Pr\{X_k^d < m\} = Pr\{X_k^s < m\} \qquad \textit{for } 1 \leq m < T.$$

and

$$Pr\{X_k^d = T\} \geq Pr\{X_k^s = T\}$$

The following proposition establishes some insightful relationships between the moments of interdeparture and the background traffic process.

Proposition 8 : *Under overload conditions at the scheduler the following can be shown :*

$$\begin{aligned} &(a) \quad E\{X_k^s\} = T * E\{A^1\} + 1 = T * (N-1) * p_b + 1 \\ &(b) \quad VAR\{X_k^s\} = T * VAR\{A^1\} = T * (N-1) * p_b * (1-p_b) \\ &(c) \quad E\{X_k^d\} = E\{W\} * E\{A^1\} + 1 = E\{W\} * (N-1) * p_b + 1 \end{aligned}$$

Where the last part of the above equations is derived for a binomial random variable A^1 with maximum value $N-1$ and success probability p_b; $VAR\{x\}$ denotes the variance of random variable x.

Corollary 2 : *The throughput gain of the* dynamic-R&S *scheme over that of the* static-R&S *scheme under overload conditions increases as the load increases and its asymptotic value is equal to T. That is :*

$$\frac{1/E\{X_k^d\}}{1/E\{X_k^s\}} = \frac{T + 1/E\{A^1\}}{E\{W\} + 1/E\{A^1\}} \xrightarrow{E\{A^1\}\to\infty} T$$

6 Numerical Results

In this section, some numerical results are presented to quantify the behavior of the dynamic-R&S and static-R&S schemes. The results are derived under heavy traffic source load at the regulator and both underload and overload conditions at the scheduler. Although the heavy traffic source and overload conditions are not the dominant ones in a well designed system, they will be present if substantial statistical multiplexing gain is to be achieved. And while simple regulator schemes - such as the static-R&S one - may be adequate under nominal (underload) traffic conditions, it is important that their behavior under less frequent - but QoS compromising - overload conditions, be investigated. The results shown here indicate that the dynamic-R&S scheme proposed in this paper outperforms the static-R&S and, importantly, can cope with overload conditions better than the static-R&S one.

The results presented below have been obtained for two values of the target interdeparture time T - or desirable throughput $1/T$ - equal to $T = 5$ and $T = 10$. The background traffic is modeled as an independent per slot, batch process with binomially distributed batch size of maximum value $N - 1 = 8$ and success probability p_b. The background load or utilization is denoted by ρ_{back}. In a system were all - or a large subset - of the connections are regulated, the background process will not be independent per slot. This assumption is made for analysis tractability, since modeling the correlation in the background would make the analysis of the system impossible. Also the above assumption is not completely unrealistic, since in each slot a number of eligibility times will mature and a batch of cells will be delivered to the scheduler.

Fig. 6 presents the regulator throughput vs ρ_{back} under both policies, obtained from Proposition 2. As ρ_{back} increases, the dynamic-R&S policy can detect the increased background intensity and release cells earlier, attempting to provide the targeted throughput $(1/T)$ and control jitter. As a consequence, the rate by which the packets leave the regulator increases as the background intensity increases and it will reach a maximum of 1 if at least $T - 1$ background cells are delivered to the scheduler in each slot (high overload at the scheduler). The scheduler throughput S_{max} vs ρ_{back} is shown in Fig. 7 for both policies under underload conditions $(\rho < 1)$. S_{max} is calculated as $1/E\{X_k\}$; the Probability Mass Function (PMF) of X_k is calculated from (16) for the dynamic policy, and from (17) for the static policy. As expected from (12), $S^d_{max} = R^d_{max}$ under underload conditions and infinite buffer capacity at the scheduler. Although R^d_{max} increases to 1 as ρ_{back} increases (as said earlier), S^d_{max} starts deviating from R^d_{max} and declines beyond some value of ρ_{back} equal to about .78 for $T = 5$. This is due to the fact that the scheduler reaches an overload state $(\rho > 1)$ and the dynamics change. Specifically, an infinite scheduler queue backlog is built up under these conditions, resulting in a different than R_{max} output rate from the scheduler, as discussed earlier. Thus, only the results for $R_{max} + \rho_{back} < 1$ are relevant. The scheduler study under overload should be employed for the derivation of results for $\rho > 1$.

Results for S_{max} vs ρ_{back} are shown in Fig. 8 for ρ_{back} such that the scheduler is in overload state ($\rho > 1$). The results can be calculated either from parts (a) and (c) of Proposition 8, or from the PMF of X_k derived in (19) and (18) The improved throughput characteristics of the dynamic-R&S scheme can be clearly observed. The higher than the targeted throughput for low overload conditions is in accordance with expectations based on the increased regulation throughput and the heavy traffic assumption for the tagged source. As long as $\rho < 1$, the regulator throughput determines the throughput at the scheduler as well, as explained earlier. When $\rho > 1$, some of the regulator traffic is "absorbed" by the infinite buffer built up at the scheduler and a throughput reduction is observed for this reason. Nevertheless, by reducing W (see equation (1), the dynamic-R&S scheme is capable of providing the targeted throughput under severe overload conditions. In the limiting case of very large overload, $W \rightarrow 1$ and the tagged throughput reduction below the targeted value is observed, induced by the per slot background batch size. It should be noted that under the static-R&S scheme the tagged throughput falls dramatically even under low overload conditions. This reduction is directly related to the cumulative over T slots background arrivals, as opposed to that over W slots ($W \rightarrow 1$) under the dynamic-R&S scheme.

In addition to maintaining a throughput as close to the targeted one under overload, it is important that the variability of the throughput (or interdeparture time X_k) be low. Fig. 9 presents results for the variance of the tagged cell interdeparture process induced by the two policies vs ρ_{back} and for underload conditions at the scheduler ($\rho < 1$). Again, only the results for $R_{max} + \rho_{back} < 1$ are relevant. It is clear that the dynamic-R&S policy provides for a less variable interdeparture process than the static-R&S one.

Similar results for $\rho > 1$ are presented in Fig. 10. These results have been derived by employing the PMF of X_k derived in Proposition 6, for various values of ρ_{back} corresponding to success probabilities given by $p_k = 0.12 + (k-1) * 0.02$ for $k = 1, 2, \ldots 13$. In view of the linear relationship between $VAR(X_k)$ and $VAR(A^1)$ under the static-R&S scheme (Proposition 8) the increasing behavior of $VAR(X_k)$ as ρ_{back} increases is expected and it is observed in Fig. 10. The results under the dynamic-R&S scheme are more difficult to interpret. For low overload conditions $VAR(X_k)$ decreases until $\rho_{back} = 1.44$ (4th point on the plot) and then increases slightly. X_k depends solely on the background accumulation over W slots (between consecutive tagged cell releases). Therefore, as ρ_{back} increases the condition $A^W \geq T - 1$ is expected to be met in fewer slots, and therefore a cell would be released earlier. This implies that a decreasing number of batches would interfere with X_k, reducing $VAR(X_k)$. As ρ_{back} increases the number of batches which interfere with X_k under the dynamic-R&S policy reduces to 1; beyond that point the increased value of $VAR(X_k)$ is due to the increase in $VAR(A^1)$.

The detailed traffic smoothness characteristics of the two R&S schemes are shown in Fig. 11 - 16 for underload and Fig. 17 - 20 for overload conditions at the scheduler, where the PMF of X_k (jitter) and its tails are plotted.

The jitter (X_k) PMF under both policies is plotted in Fig. 11 for $\rho_{back} = .45$ and $T = 5$ and $T = 10$. Since $\rho << 1$ the two policies behave similarly and the tails of the PMF almost coincide. For higher background utilization ($\rho_{back} = .85$) the PMF of X_k becomes quite distinct for the two policies (Fig. 12). The induced clustering ($X_k < T$) and spreading ($X_k > T$) are better shown in terms of the tails of the PMF presented in Fig. 13 (15) and 14 (16), respectively, for $T = 5$ ($T = 10$). Tagged cell clustering ($X_k < T$) is seen to slightly increase under the dynamic-R&S policy while spreading ($X_k > T$) is substantially reduced and more probability mass is concentrated around T. While the spreading reduction under the dynamic-R&S policy is expected, the slight increase in the clustering is less obvious. It may be attributed to the higher probability that the scheduler queue is nonempty under the dynamic-R&S policy, due to the higher scheduler load (S_{max}) resulting from a higher regulator throughput (R_{max}). This is discussed also below.

The traffic smoothness characteristics of the two schemes under overload traffic conditions at the scheduler can be observed in Fig. 17 and 18, where the jitter distribution (or scheduler interdeparture distribution) is plotted for two different values of ρ_{back} (Back Util). In both cases the jitter probability mass function is highly contained around the target value T under the dynamic-R&S scheme, and it is spread over wide range of values under the static-R&S scheme.

The tails of the probability mass function representing spreading ($Pr\{X_k \geq k\}$ for $k > T$), are shown in Fig. 19 and 20 for $T = 5$ and $T = 10$, respectively. The good jitter characteristics of the dynamic-R&S scheme in terms of reduced cell spreading are clearly observed. Excessive spreading – occurring under network congestion (scheduler overload) – may compromise the QoS of a real-time application by causing starvation at the end user. It is evident that the starvation probability can be substantially lower under the dynamic-R&S scheme. As indicated in Proposition 7 the policies induce identical clustering since the latter depends *only* on the number of background arrivals over T slots under overload conditions. Under underload conditions, the empty scheduler queue process also modulates clustering. Since this process is different under the two policies, different clustering is expected under underload conditions at the scheduler (shown in Fig. 15).

7 Simulation Results

7.1 The simulated system

In the preceding sections the R&S policies were considered for a single (tagged) source. All other co-existing sources were aggregated and formed the background source. This consideration was necessary in order to analytically study the behavior of the system and gain insight on the operation of the R&S policies.

In this section, a system in which more than one sources are controlled by the R&S policies is considered and is studied by employing the OPtimized NEtwork simulation Tool (OPNET). The objective here is to investigate the behavior of the two policies in the presence of real background traffic, as generated by multiple sources controlled by these policies. The numerical results presented in this section are consistent with the expectations shaped by the analysis.

A system with $N = 7$ ON_OFF Markov sources was simulated. Such sources are defined in terms of the parameters of the geometric distributions governing the lengths of the ON (P_{on}) and OFF (P_{off}) periods and the cell generation rate $Parr$ when the source is in the active (ON) state (no cells are generated when the source is in the inactive (OFF) state). The empirical PMF and the variance of the interdeparture process and the scheduler throughput for each of the sources were measured. The actual cell interdeparture times (X_k) from the scheduler were recorded, after filtering out the gaps caused by the source's OFF periods, and a vector $\bar{X} = [X_1 X_2 \dots X_k \dots]$ was created for each of the sources. The empirical interdeparture PMF for each source was obtained as the histogram of the samples in $\bar{X}$. The per source throughput was calculated as the inverse of the average of the samples in $\bar{X}$. Since $\bar{X}$ does not contain the OFF periods, the calculated throughput will be higher than the actual average throughput. This calculated throughput is called the active throughput, S_a. This filtering of the data allows for capturing the effect that the policies have on the sources while they are active, without being obscured by the OFF periods where the policies are ineffective. S_a can be viewed as the average rate of service provided to a source while it is in the ON state. The variance of the inter departure time is expressed in terms of a measure of its deviation from the target spacing $X_{min,i}$ and is calculated as $\frac{1}{L-1}\sum_{k=1}^{L}(X_k - X_{min,i})^2$, for source i and sample size of L.

7.2 Results

A system with $N = 7$ Markov sources was considered in which more than one sources are controlled by the R&S policies. The parameters and results of the first simulation are shown in Table 1; $P_{on} = P_{off} = .8$ for src_0, src_1, src_2, $P_{on} = .8$ and $P_{off} = .85$ for src_3, src_4 and $P_{on} = .8$ and $P_{off} = .75$ for src_5, src_6. For each source, the value of X_{min} and the measured average source rate λ_{ave} are shown, along with the measured active throughput S_a^d and variance $VAR[X_k^d]$ under the dynamic-R&S and S_a^s and variance $VAR[X_k^s]$ under the static-R&S policies.

The Statistical Multiplexing Gain (SMG) – defined as $\sum_i 1/X_{min,i}$ – is equal to 3.6 in this case. The system utilization, ρ, is equal to 0.9505 ($\rho = \sum_i \lambda_{ave,i}$), which should be less than 1 for a stable system. The sources with $X_{min} = 1$ are practically unregulated since they are allowed to release their cells to the scheduler as soon as they are generated. Such sources could be ones without jitter constraints. Sources with $X_{min} > 1$ are the ones targeted for regulation; for these sources, X_{min} is set to $\lfloor 1/Parr \rfloor$.

The positive impact of the dynamic-R&S policy on the regulated sources ($src_3, src_4, src_5, src_6$) is clearly observed in Table 1: the variance of the interdeparture process under the dynamic-R&S scheme ($VAR[X_k^d]$) is substantially smaller than the variance under the static-R&S scheme ($VAR[X_k^s]$). This

src #	X_{min}	λ_{ave}	S_a^d	S_a^s	$VAR[X_k^d]$	$VAR[X_k^s]$
src_0	1	0.2470	0.44860	0.45551	3.2604	3.0234
src_1	1	0.1952	0.39611	0.40240	4.8649	4.6036
src_2	1	0.2004	0.40216	0.40867	4.6249	4.3924
src_3	10	0.0434	0.10768	0.09611	4.6240	9.3360
src_4	10	0.0422	0.10568	0.09444	4.0288	8.8864
src_5	5	0.1105	0.20280	0.18248	3.9422	5.1855
src_6	5	0.1118	0.20731	0.18620	3.3056	4.5055
	$\sum_i 1/X_{min,i} = 3.6$	$\sum_i \lambda_{ave,i} = 0.9505$				

Table 1: Simulation parameters for the 1^{st} simulation scenario

quantifies the ability of the dynamic-R&S scheme to control the delay jitter better than the static-R&S scheme, for all of the regulated sources. As a result, under the dynamic-R&S scheme, S_a^d reaches the target value ($1/X_{min}$) and exceeds it slightly, due to some residual clustering which is present under both policies. On the other hand, under the static-R&S scheme S_a^s is lower than the target value due to the static nature of the policy. The dynamic-R&S scheme manages to serve the regulated sources with a less variable service rate, which reaches the target peak service rate. As a result, the traffic is better shaped at the output of the scheduler, producing the desired performance.

The unregulated sources (src_0, src_1, src_2) experience a slightly higher $VAR[X_k^d]$. Since the target X_{min} for these sources is 1, the two policies are basically ineffective and the variance and active throughput are shaped solely by the background interfering process seen by each of these sources during each slot. The background interfering processes are different under the two policies. It is expected that under the dynamic-R&S policy more cells will be released to the scheduler per slot during the ON period, due to the early releases. Therefore, more background traffic interferes with the unregulated sources under the dynamic-R&S scheme resulting in more spreading which lowers S_a^d.

The above results can be viewed graphically in terms of the empirical PMFs shown in Fig. 21 and Fig. 22. The probability mass at the target value is substantially higher under the dynamic-R&S scheme. This may be attributed to the reduced spreading. Every time a cell is released earlier than the release time under the static-R&S policy, potential spreading is avoided. The empirical PMFs for $X_k < X_{min}$ almost coincide under the two policies, implying that the two policies generate the same amount of clustering. The empirical PMFs for src_0, src_1, src_2 are almost identical under the two policies and they are not shown since they do not provide any insight.

In order to study the system under extreme overload, the average rate of the three unregulated sources was increased (Table 2); the values of P_{on} and P_{off} are the same as in the first simulation. This could reflect a scenario according to which the unregulated sources start to misbehave and become

src #	X_{min}	λ_{ave}	S_a^d	S_a^s	$VAR[X_k^d]$	$VAR[X_k^s]$
src_0	1	0.39766	0.43032	0.43507	3.1226	2.9486
src_1	1	0.38366	0.42681	0.43153	3.1327	2.9804
src_2	1	0.40030	0.43270	0.43707	3.0266	2.9132
src_3	10	0.04330	0.09293	0.06548	8.2622	48.0283
src_4	10	0.04230	0.09089	0.06389	8.9868	51.9631
src_5	5	0.11112	0.15797	0.12405	11.995	20.3197
src_6	5	0.11180	0.16287	0.12627	9.8256	17.7105
	$\sum_i 1/X_{min,i} = 3.6$	$\sum_i \lambda_{ave,i} = 1.4905$				

Table 2: Simulation parameters for the 2^{nd} simulation scenario

bandwidth greedy. The utilization of the system was brought up to 1.4905 for some time and then the sources were turned off and the scheduler was served until it became empty. This way, the behavior of the two policies under temporary overload conditions could be studied.

Table 2 shows results for the variance of the interdeparture process and active throughput under this scenario. The variance of the regulated sources under the static-R&S policy increases dramatically. This is not the case under the dynamic-R&S policy which manages to keep the variance substantially lower. Even though under extreme overload the analytical results predict that S_a^d should be fairly close to the target value $1/X_{min}$ (Fig. 8), it is not seen here. The latter is due to the heavy traffic assumption made in the analytical study, under which the regulator never empties and the maximum effect of the policy can be revealed. In the simulations, the regulator can be empty when the conditions for a cell release are met. The eligible cell will be released in a later slot as soon as it arrives, allowing the misbehaving sources to secure more bandwidth. The empirical PMFs show a dramatic spreading of the interdeparture time under the static R&S scheme. The dynamic-R&S scheme manages to highly contain the PMF around the target value, which is the desired behavior. Due to space limitation these plots are not presented.

In order to study the system under statistical multiplexing gain of less than 1, the simulation with the parameters shown in Table 3 was considered; $SMG = 0.93$ and $\rho = 0.7287$; $P_{on} = .9$ and $P_{off} = .2$ for $src_0, src_1, src_2, src_3$, $P_{on} = .9$ and $P_{off} = .4$ for src_4, src_5 and $P_{on} = .8$ and $P_{off} = .75$ for src_6. The results show that the dynamic-R&S policy is now outperformed by its static counterpart. $VAR[X_k^d]$ and S_a^d are both higher. Under such circumstances a higher S_a^d cannot be considered to be an improvement, since it is due to more clustering. Since $SMG < 1$, there are no periods of overload, but the conditions for early release can be met for some of the sources. As cells are released earlier the scheduler occupancy under the dynamic-R&S scheme is expected to be higher, and two consecutive cells can be clustered. Under the static-R&S scheme the scheduler occupancy is lower (below 7) resulting

src #	X_{min}	λ_{ave}	S_a^d	S_a^s	$VAR[X_k^d]$	$VAR[X_k^s]$
src_0	10	0.09970	0.09979	0.09914	1.7742	1.5668
src_1	10	0.09949	0.09949	0.09899	2.3031	1.8366
src_2	10	0.09911	0.09911	0.09845	3.3517	3.0075
src_3	10	0.09987	0.09987	0.09883	2.4080	2.1334
src_4	6	0.16314	0.16314	0.15813	4.8881	3.7537
src_5	6	0.16377	0.16377	0.15878	4.0384	2.9886
src_6	5	0.20161	0.20161	0.19424	2.6167	1.6143
	$\sum_i 1/X_{min,i} = 0.93$	$\sum_i \lambda_{ave,i} = 0.7287$				

Table 3: Simulation parameters for the 3^{rd} simulation scenario

in lower clustering.

In order to run this simulation with $SMG = 0.93$ and $\rho = 0.74$ the arrival rates of the sources were increased by reducing the length of the OFF and increasing the length of the ON periods. This makes the sources less bursty and it makes peak rate allocation of connections cost efficient. In a highly utilized network this scenario is not expected to be the dominant one. Instead the previous two are expected to be the dominant ones, under which the dynamic-R&S scheme clearly outperforms the static-R&S scheme.

A Proofs of Propositions and Corollaries

This appendix contains al the proofs for the propositions and corollaries presented in the paper.

A.1

Proof of Proposition 1

Proof: Equation (8) is obtained by noting that the events $\{W \geq j\}$ and $\{\sum_{i=1}^{j-1} A_i \leq T-2\}$ are equivalent for $1 < j < T$, writing

$$\begin{aligned} Pr\{W=j\} &= Pr\{W \geq j\} - Pr\{W \geq j+1\} = Pr\{\sum_{i=1}^{j-1} A_i \leq T-2\} - Pr\{\sum_{i=1}^{j} A_i \leq T-2\}, \\ &= \tilde{F}_{j-1}(T-2) - \tilde{F}_j(T-2) \end{aligned}$$

and noting that

$$\begin{aligned} & Pr\{W=1\} = Pr\{A^1 \geq T-1\} = \\ & = 1 - Pr\{A^1 \leq T-2\} = \tilde{F}_0(T-2) - \tilde{F}_1(T-2) \end{aligned} \tag{20}$$

and

$$Pr\{W=T\} = Pr\{\sum_{i=1}^{T-1} A_i \le T-2\} = \tilde{F}_{T-1}(T-2) - \tilde{F}_T(T-2) \tag{21}$$

Equation (9) can be shown by considering the indicator function I_j defined as follows. Let

$$I_j = \begin{cases} 1 & \text{if the previous } (j-1) \text{ batches are such that } \sum_{i=1}^{j-1} A_i \le T-2 \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

for $1 < j \le T$, and let $I_1 \doteq 1$. Notice that $I_j = 1$ implies that $W \ge j$ and, thus,

$$W = \sum_{j=1}^{T} I_j \tag{23}$$

Application of the expectation operator to the above leads to the following.

$$E\{W\} = E\{\sum_{j=1}^{T} I_j\} = \sum_{j=1}^{T} E\{I_j\} = \sum_{j=2}^{T} Pr\{\sum_{i=1}^{j-1} A_i \le T-2\} + 1 = \sum_{j=1}^{T} \tilde{F}_{j-1}(T-2) \ \square \tag{24}$$

A.2

Proof of Proposition 2

Proof: Equations (10) and (11) are self-evident in view of the heavy traffic source assumption which is typically made to establish the maximum throughput and in view of (9). The inequality can be proven by noting that $W = min\{B,T\} \le T$ with equality only if the stated condition is met. Alternatively, notice that $T = \sum_{j=1}^{T} 1 \ge \sum_{j=1}^{T} \tilde{F}_{j-1}(T-2) = E\{W\}$ since $\tilde{F}_{j-1}(T-2) \le 1$; equality holds if $\tilde{F}_{j-1}(T-2) = 1$ or $Pr\{\sum_{i=1}^{j-1} A_i \ge T-1\} = 0$ for $1 \le j \le T$ or, equivalently, if $Pr\{\sum_{i=1}^{T-1} A_i \ge T-1\} = 0$ which is the stated condition. □

A.3

Proof of Proposition 3

Proof: The scheduler interdeparture process X_k can be determined by the scheduler occupancy Q_k found upon arrival of the k^{th} tagged cell (not counting itself). Since the next arrival to the scheduler will occur after $W_k = min\{B_k, T\}$ slots, the scheduler can serve at most T cells until the next tagged cell arrival occurs. Two cases need to be distinguished, depending on whether $Q_k \ge T-1$ or $Q_k \le T-2$.

Case I: $Q_k \ge T-1$

In this case X_k is equal to the number of background cells which arrived over W_k plus 1 slots (see Fig 4).

Case I.1: $0 \le l \le N-2$ (no-clustering)

Let m, $1 \le m \le T$, be the slot over which the $(k+1)$st tagged cell enters the scheduler queue, following the arrival of the k^{th} tagged cell, that is, $W_k = m$. Clearly, $A^m = X_k - 1 = T + l - 1$ and

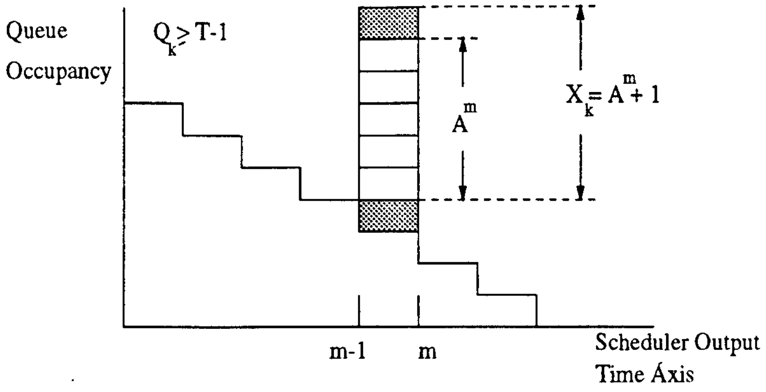


Figure 4: Illustration of $X_k = A^m + 1$ when $W_k = m$ and $Q_k \geq T - 1$

$A^{m-1} \leq T - 2$. The smallest possible value of X_k not inducing clustering is $X_k = T$. On the other hand, the largest possible value of X_k allowed by the dynamic-R&S policy is $X_k = T + N - 2$, since it is possible that $T - 2$ background cells have been accumulated by the $(m - 1)$st slot and a batch of maximum size (equal to $N - 1$) occurs during slot m. As a result, $T \leq X_k \leq T - 2 + N - 1 + 1$ implying that $0 \leq l \leq N - 2$. In this case,

$$Pr\{X_k = T + l/Q_k \geq T - 1\} = \sum_{m=1}^{T} Pr\{A^m = T + l - 1, A^{m-1} \leq T - 2\}$$

Case I.2: $-(T-1) \leq l < 0$ (clustering)

Clustering can happen only if the background arrivals over T slots do not exceed $T - 2$. At slot T a cell will be released to the scheduler ($V_k = T$), creating clustering. Since $Q_k \geq T - 1$ it is implied that $X_k = A^T + 1$ and, thus,

$$Pr\{X_k = T + l\} = Pr\{A^T = T + l - 1\}.$$

Case II: $Q_k = i \leq T - 2$

Since $Q_k = i \leq T - 2$, it is possible that the server will exhaust the $Q_k + 1$ cells before the $(k+1)$st tagged cell arrives and serve some of the background built up behind the k^{th} cell (see Fig. 5). As in Case I, the following cases are considered.

Case II.1: $l \geq 0$ (no-clustering)

Let m be the slot over which the $(k+1)$st tagged cell enters the scheduler, following the arrival of the k^{th} tagged cell (that is, $W_k = m$).

Case II.1-(i) $1 \leq m \leq i + 1$

In this case the server has not had a chance to serve any of the background cells built up behind the k^{th} tagged cell. This case is then equivalent to Case I.1 and therefore,

$$Pr\{X_k = T + l/Q_k = i\} = \sum_{m=1}^{i+1} Pr\{A^m = T + l - 1, A^{m-1} \leq T - 2\}.$$

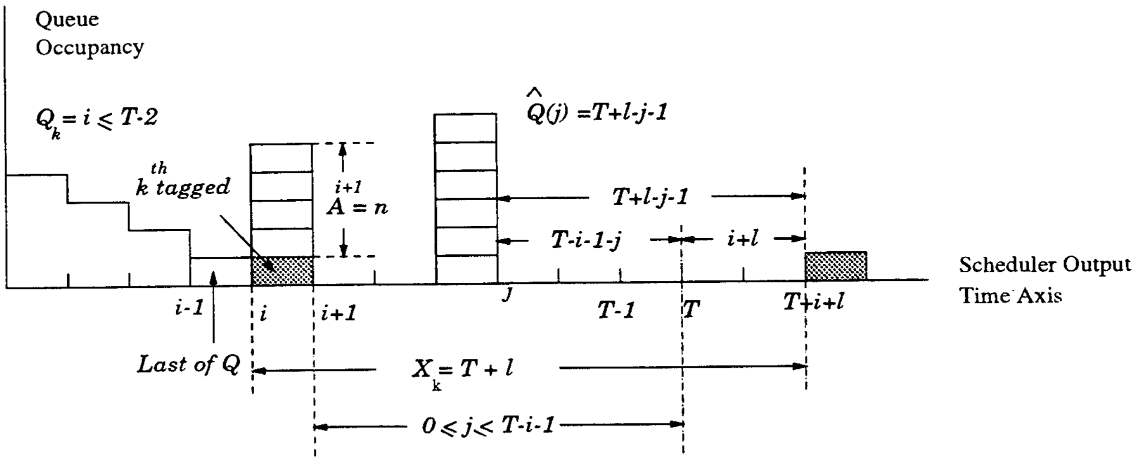


Figure 5: $V_k = j$, $Q_k \leq T - 2$: Scheduler can serve up to j cells form the A^{i+1}

Case II.1-(ii) $i + 2 \leq m \leq T$

In this case the server has served the k^{th} tagged cell, as well as some of the background built up behind the k^{th} tagged cell, before slot m. X_k will be determined by the scheduler occupancy at slot m. Since time slot m depends on the background accumulation behind the kth tagged cell, and this quantity cannot be determined by the occupancy process, both the background accumulation process behind the kth tagged cell and the queue occupancy process should be considered jointly after time $i + 1$. (see Fig. 5).

Let $\hat{A}(j)$ denote the cumulative background arrivals between time slot $j = 0$ (which is slot $i + 1$) and (including) slot $j = T - i - 1$, following slot $i + 1$. Let $\hat{Q}(j)$ denote the scheduler queue occupancy at the j^{th} slot, including arrivals during the j^{th} slot. The following lemma describes the evolution of process $\{\hat{A}(j), \hat{Q}(j)\}_{j=0}^{T-i-1}$.

Lemma 1 : *The evolution of process* $\{\hat{A}(j), \hat{Q}(j)\}_{j=0}^{T-i-1}$ *is given by*

$$\hat{A}(j) = \hat{A}(j-1) + A^1 \tag{25}$$

$$\hat{Q}(j) = max\{\hat{Q}(j-1) - 1, 0\} + A^1 \tag{26}$$

and, thus, the process evolves as a Markov chain. The initial state of this process will be taken to be given by $\{\hat{A}(0), \hat{Q}(0)\} = \{0, n\}$, *where* n *is the queue occupancy at slot* $i + 1$. *The 1-step transition probabilities are given by:*

$$Pr\{\hat{A}(j) = i_2, \hat{Q}(j) = j_2 / \hat{A}(j-1) = i_1, \hat{Q}(j-1) = j_1\} =$$
$$= \begin{cases} Pr\{A^1 = i_2 - i_1\} & if\, j_2 = max\{j_1 - 1, 0\} + i_2 - i_1 \\ 0 & otherwise. \end{cases}$$

Since the evolution of this process will be terminated when the cumulative background arrivals $n + \hat{A}(j)$ *exceed* $T - 2$, *the k-step-constrained (by* $n + \hat{A}(j) \leq T - 2$*) evolution of* $\{\hat{A}(j), \hat{Q}(j)\}_{j=0}^{j=T-i-1}$ *will be considered only. The corresponding k-step-constrained transition probabilities are given by the following*

recursions, where
$Pr^k\{\hat{A}(j) = a_F, \hat{Q}(j) = q_F/\hat{A}(j-k) = a_I, \hat{Q}(j-k) = q_I\} = Pr^k\{a_F, q_F/a_I, q_I\}$,

$$\begin{aligned}
&Pr^k\{a_F, q_F/a_I, q_I\} = \\
&\quad = \sum_{a_1=0}^{T-2-q_I} Pr\{a_1, q_1/a_I, q_I\} * Pr^{k-1}\{a_F, q_F/a_1, q_1\} = \\
&\quad \mathit{where} \quad q_1 = max\{q_I - 1, 0\} + a_1 - a_I \\
&\quad = \sum_{a_1=0}^{T-2-q_I} \sum_{a_2=a_1}^{T-2-q_I} Pr\{a_1, q_1/a_I, q_I\} * Pr\{a_2, q_2/a_1, q_1\} * Pr^{k-2}\{a_F, q_F/a_2, q_2\} = \cdots \\
&\quad \mathit{where} \quad q_2 = max\{q_1 - 1, 0\} + a_2 - a_1 \quad \Box
\end{aligned}$$

As it can be seen in Fig. 5, $\hat{Q}(j) = T + l - j - 1$ in order to achieve a target $X_k = T + l$, given that the $(k+1)$st cell arrives at the scheduler during the j^{th} slot, $1 \leq j \leq T - i - 2$. At the same time $\hat{A}(j) \geq T - 1 - n$, since $\hat{A}(j)$ is equal to the background accumulation *after* the $(i+1)$st slot and at slot $i+1$ there were $\hat{Q}(0) = n = A^{i+1}$ background cells behind the k^{th} tagged cell. Given $\{\hat{A}(0) = 0, \hat{Q}(0) = n\}$ the event that will provide for the target $X_k = T + l$ is $\{\hat{A}(j) \geq T - 1 - n, Q(j) = T + l - j - 1\}$. This event can happen any time between $1 \leq j \leq T - i - 2$; its probability is given by the j-step-constrained transition probability of the two dimensional Markov chain $\{\hat{A}(j), \hat{Q}(j)\}_{j=0}^{T-i-1}$ which was derived earlier. The slot $j = T - i - 1$ is considered differently, since at $j = T - i - 1$ or $m = T$ a cell will be released independently of the state of $\hat{A}(T - i - 1)$. In this case, the event of achieving the desired $X_k = T + l$ is given as $\{\hat{A}(T - i - 1) \geq 0,\ \hat{Q}(T - i - 1) = i + l\}$ (again, the constrained evolution of $\{\hat{A}(j), \hat{Q}(j)\}_{j=0}^{j=T-i-1}$ is considered.

In view of the above,

$$\begin{aligned}
&Pr\{X_k = T + l/Q_k = i\} = \sum_{n=0}^{T-2} Pr\{A^{i+1} = n, X_k = T + l\} \\
&= \sum_{n=0}^{T-2} Pr\{X_k = T + l/A^{i+1} = n\} * Pr\{A^{i+1} = n\} \\
&= \sum_{n=0}^{T-2} Pr\{X_k = T + l/\hat{A}(0) = 0, \hat{Q}(0) = n\} * Pr\{A^{i+1} = n\} \\
&= \sum_{n=0}^{T-2} \sum_{j=1}^{T-i-2} \sum_{a_j=T-1-n}^{T-3+N-n} Pr^j\{\hat{Q}(j) = T + l - j - 1, \hat{A}(j) = a_j/\hat{A}(0) = 0, \hat{Q}(0) = n\} * \\
&\quad * Pr\{A^{i+1} = n\} + \\
&+ \sum_{n=0}^{T-2} \sum_{a_j=0}^{T-3+N-n} Pr^{T-i-1}\{\hat{Q}(j) = i + l, \hat{A}(j) = a_j/\hat{A}(0) = 0, \hat{Q}(0) = n\} * Pr\{A^{i+1} = n\}
\end{aligned}$$

<u>Case II.2:</u> $l < 0$ (clustering)

Clustering can happen only if the background accumulation over T slots is less than $T - 1$ cells. This can be seen in Fig. 5, by setting $j = T - i - 1$, which gives $\hat{Q}(T - i - 1) = i + l$, and since

$l < 0$, this creates clustering. By employing similar argument as in Case II.1 it can be shown that given $\{\hat{A}(0) = 0, \hat{Q}(0) = n\}$, the event that will provide for the target $X_k = T + l$ is $\{\hat{A}(T-i-1) \leq T-2-n,\ \hat{Q}(T-i-1) = i+l\}$. Thus,

$$Pr\{X_k = T+l/Q_k = i\} = \sum_{n=0}^{T-2} Pr\{A^{i+1} = n, X_k = T+l\}$$

$$= \sum_{n=0}^{T-2} \sum_{a=0}^{T-2-n} Pr^{T-i-1}\{\hat{A}(T-i-1) = a, \hat{Q}(T-i-1) = i+l/\hat{A}(0) = 0, \hat{Q}(0) = n\} *$$
$$* Pr\{A^{i+1} = n\}$$

Putting all the cases together employing indicator functions, equations (13) and (14) are obtained.

□

A.4

Proof of Proposition 4

Proof: Let Q_k be the scheduler queue occupancy upon arrival of the k^{th} tagged cell (without counting itself) and let m denote the slot over which the $(k+1)$st tagged cell arrives. The following cases need to be considered.

Case I: $Q_k = i \geq T-1$

It is easy to establish that $\{Q_k\}_{k \geq 1}$ for $Q_k = i \geq T-1$ evolves as a Markov chain according to the equations:

$$Q_{k+1} = \begin{cases} Q_k + 1 - m + A^m & \text{for } 1 \leq m \leq T-1 \\ & \text{if } T-1 \leq A^m \leq T-3+N,\ A^{m-1} \leq T-2 \\ Q_k + 1 - T + A^T & \text{for } m = T \\ & \text{if } A^{T-1} \leq T-2 \end{cases} \tag{27}$$

and the transition probabilities are easily determined as follows:

$$Pr\{Q_{k+1} = j/Q_k = i\} =$$
$$= Pr\{i+1-m+A^m = j, T-1 \leq A^m \leq T-3+N, A^{m-1} \leq T-2, 1 \leq m \leq T-1\} +$$
$$+ Pr\{i+1-T+A^T = j, A^T \leq T-3+N, A^{T-1} \leq T-2\} =$$
$$= Pr\{A^m = j-i-1+m, T-1 \leq A^m \leq T-3+N, A^{m-1} \leq T-2, 1 \leq m \leq T-1\} +$$
$$+ Pr\{A^T = j-i-1+T, A^T \leq T-3+N, A^{T-1} \leq T-2\} =$$
$$= Pr\{A^1 = j-i\} * 1_{\{T-1 \leq j-i \leq N-1\} \cap \{T-1 \leq N-1\}} +$$
$$+ \sum_{m=2}^{T-1} Pr\{A^m = j-i+m-1, A^{m-1} \leq T-2\} * 1_{\{T-m \leq j-i \leq T-2+N-m\} \cap \{T-1 \leq m*(N-1)\}} +$$
$$+ Pr\{A^T = j-i-1+T, A^{T-1} \leq T-2\} * 1_{\{-(T-1) \leq j-i \leq N-2\}}$$

Case II: $Q_k = i \leq T-2$

A typical evolution of the scheduler queue when $Q_k \leq T-2$ is shown in Fig. 5. A tagged cell can arrive either during the first $i+1$ slots, or during the interval $[i+2, T]$. In the former case the queue evolution is described similarly to the case described above for $Q_k \geq T-1$. In the later case, the queue evolution is described by employing the Markov chain defined in Lemma 1. The system equations are given by :

$$Q_{k+1} = \begin{cases} Q_k + 1 - m + A^m & \text{for } 1 \leq m \leq i+1 \\ & \text{if } T-1 \leq A^m \leq T-3+N \text{ and } A^{m-1} \leq T-2 \\ \hat{Q}(n) & \text{for } i+2 \leq m \leq T-1 \text{ and } 1 \leq n \leq T-i-2 \\ & \text{if } \hat{A}(n) \geq T-1-A^{i+1} \text{ and } \hat{A}(n-1) \leq T-2-A^{i+1} \\ \hat{Q}(T-i-1) & \text{for } m = T \\ & \text{if } \hat{A}(T-i-2) \leq T-1-A^{i+1} \end{cases} \tag{28}$$

It is easy to establish that $\{Q_k\}_{k \geq 1}$ for $Q_k = i \leq T-1$ evolves as a Markov chain. Its transition probabilities are easily as follows:

Transition Probabilities

For $1 \leq l \leq i+1$ the transition probabilities can be obtained by the same reasoning as in Proposition 3 :

$$Pr\{Q_{k+1} = j/Q_k = i\} =$$

$$Pr\{A^1 = j-i\} * 1_{\{T-1 \leq j-i \leq N-1\} \cap \{T-1 \leq N-1\}} +$$

$$+ \sum_{m=2}^{i+1} Pr\{A^m = j-i-1+m, A^{m-1} \leq T-2\} * 1_{\{T-m \leq j-i \leq T-2+N-m\} \cap \{T-1 \leq m*(N-1))\}}$$

Note the upper bound on the index of summation.

For $i+2 \leq l \leq T-1$ the $Pr\{Q_{k+1} = j/Q_k = i\}$ when $A^{i+1} = n$ (see Fig. 5), can expressed in terms of $\{\hat{A}(n), \hat{Q}(n)\}$ as follows :

$$Pr\{Q_{k+1} = j/Q_k = i\} =$$

$$Pr^j\{\hat{A}(j) = a_j, \hat{Q}(j) = j/\hat{A}(0) = 0, \hat{Q}(0) = n\} * Pr\{A^{i+1} = n, A^i \leq T-2\}$$

for $0 \leq n \leq T-2$

and $1 \leq j \leq T-i-2$

and $T-1-n \leq a_j \leq T-3+N-n$

The above gets translated into the triple summation of 15.

For l=T

$$Pr\{Q_{k+1} = j/Q_k = i\} =$$

$$= Pr^{T-i-1}\{\hat{A}(T-i-1) = a, \hat{Q}(T-i-1) = j/\hat{A}(0) = 0, \hat{Q}(0) = n\} *$$

$$*Pr\{A^{i+1} = n, A^i \leq T-2\}$$

for $0 \leq n \leq T-2$

and $0 \leq a \leq T-3+N-n$

The above gets translated into the double summation of 15. □

A.5

Proof of Proposition 5

Proof : Let t_k (τ_k) denote the time of the kth tagged cell arrival (departure) to (from) the scheduler queue. Then:

$$\left.\begin{aligned} \tau_k &= t_k + Q_k \\ \tau_{k+1} &= t_{k+1} + Q_{k+1} \end{aligned}\right\} \Rightarrow X_k = \tau_{k+1} - \tau_k = T + Q_{k+1} - Q_k$$

Thus,

$$Pr\{X_k = T + l\} = Pr\{Q_{k+1} - Q_k = l\} = \sum_{i=1}^{\infty} Pr\{Q_{k+1} = l + i/Q_k = i\} * \pi_q(i) \quad \square$$

A.6

Proof of Proposition 6

Proof: Under the infinite scheduler buffer and overload assumption it is evident that the scheduler will never be idle during two consecutive tagged cell interarrivals and, thus, the distribution of X_k will be given by the T-fold convolution of A^1 under the static-R&S scheme. Therefore:

$$Pr\{X_k = l\} = Pr\{A^T = l - 1\}$$

and since

$$\begin{aligned} 0 \ \leq \ & A^T \leq T * (N-1) \Rightarrow \\ 1 \leq \ & l \ \ \leq T * (N-1) + 1 \end{aligned}$$

Similarly if $W_k = m$ is the time interval between two consecutive tagged cell releases under the dynamic-R&S scheme then the condition $\{A^m = l - 1, A^m \leq T - 2\}$ must be met in order for the interdeparture to be equal to l, for $l \geq T$, where $1 \leq m \leq T$. It should be noted that the result of this proposition can be obtained directly from (13) and (14) by noting that all the indicator functions are zero since $Q_k \geq T - 1$. Therefore :

$$Pr\{X_k = T + l/Q_k \geq T - 1\} = \sum_{m=1}^{T} Pr\{A^m = T + l - 1, A^{m-1} \leq T - 2\} \qquad \text{for } l \geq 0$$

$$Pr\{X_k = T + l/Q_k \geq T - 1\} = Pr\{A^T = T + l - 1\} \qquad \text{for } l < 0$$

and

$$\begin{aligned} & Pr\{X_k = T + l\} = \\ & Pr\{X_k = T + l/Q_k \leq T - 2\} * Pr\{Q_k \leq T - 2\} + \\ & + Pr\{X_k = T + l/Q_k \geq T - 1\} * Pr\{Q_k \geq T - 1\} \\ & = Pr\{X_k = T + l/Q_k \geq T - 1\} \end{aligned}$$

Changing $T+l$ into l and with the help of indicator functions, the expression in (18) is obtained. Also since:

$$\begin{aligned} 0 &\leq A^m \leq T-3+N \Rightarrow \\ 1 &\leq l \quad \leq N+T-3 \qquad \square \end{aligned}$$

A.7

Proof op Proposition 7

Proof: Let Q_k denote the scheduler queue occupancy upon arrival of the kth tagged cell to the queue. Under the overload conditions, $Q_k \geq T-1$, and thus the kth cell will not depart before the $(k+1)$st tagged cell arrives (recall the heavy traffic assumption is made in this paper). As a result, X_k will be equal to the background arrivals over the tagged cell interarrival interval and, thus, $X_k^{co} = A_k^W - 1$ and $X_k^{no} = A^T - 1$, where W is the generic random variable for W_k and $W_k = min\{B_k, T\} \leq T$. Since $Pr\{A_k^W > m\} \leq Pr\{A_k^T > m\}$ for all m, the first claim is proved.

Clustering ($X_k < T$) occurs if and only if the arrivals of the background traffic following the arrival of the kth tagged cell to the scheduler, and before the $(k+1)$st tagged cell release, are less than $T-1$. Since under such condition the two policies operate identically, it is clear that such events will be induced by the two policies with the same probability.

The third claim follows directly from the expressions in Proposition 6 by setting $l = T$. □

A.8

Proof of Proposition 8

Proof: Parts (a) and (b) are self evident in view of the fact that under overload $X_k^{no} = \sum_{i=1}^{T} A_i^1 + 1$ and A_i^1 for $1 \leq i \leq T$, are independent and identically distributed random variables. Part (c) is also self evident in view of the fact that W is a stopping time for the random variables A_i^1, $i = 1, 2, \ldots$ and the result follows from Wald's equation (pp. 171 in [12]). □

The following interesting conclusions may be drawn from Proposition 8. This result establishes the magnitude of the difference in throughput under severe overload conditions achieved by the tagged traffic stream under the two policies.

A.9

Proof of corollary 2

Proof of corollary 2

Proof: Under very heavy traffic, $E\{A^1\} \to \infty$, the dynamic-R&S policy will release one tagged cell per slot (that is $E\{W\} \to 1$) and the above follows. □

References

[1] D. Verma D. Ferrari. "A Scheme for Real Time Channel Establishment in Wide Area Networks". *IEEE Journal on Selected Areas in Communication*, 8(3):368–379, April 1990.

[2] D. Ferrari. "Delay Jitter Control Scheme for Packet-Switching Internetwork". *Computer Communications*, 15(6):367–373, July-August 1992.

[3] D. Ferrari D. C. Verma, H. Zhang. "Delay Jitter Control for Real-Time Communication in a Packet Switching Network". In *Proceedings of Tricomm '91, Chapel Hill N.C.*, pages 35–46, April 1991.

[4] S. Keshav C. R. Kalmanek, H. Kanakia. "Rate Controlled Servers for Very High Speed Neworks". In *IEEE Global Telecommunication Conference, San Diego California*, pages 300.3.1–300.3.9, December 1990.

[5] S. J. Golestani. "Congestion-Free Communication in High-Speed Packet Networks". *IEEE Transactions In Networking*, 39(12):1802–1812, December 1991.

[6] S. Shenker A. Demers, S. Keshav. "Analysis and Simulation of a Fair Queueing Algorithm". In *Proceedings ACM SIGCOMM '89*, pages 1–12, October 1989.

[7] R. G. Galager A. K. Parekh. "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks : The Single-Node Case ". *IEEE/ACM Transactions in Networking*, 1(3):344–357, June 1993.

[8] D. Ferrari H. Zhang. "Rate-Controlled Static-Priority Queueing". In *Proceedings IEEE INFOCOM '93*, pages 227–236, September 1993.

[9] J. Pasquale N. R. Figueira. "Leave in Time: A New Service Discipline for Real Time Communications in a Packet Switching Nework". In *Proceedings ACM SIGCOMM '95 Cambridge MA*, pages 207–218, 1995.

[10] S. K. Tripathi D. Saha, S. Mukherjee. "Multi-rate Traffic Shaping and End-to-End Performance Guarantees in ATM Networks". In *International Conference on Network Protocols, 1994*, pages 188–195, 1994.

[11] L. Zhang. "Virtual Clock: A New Traffic Control Algorithm for Packet Switching Networks". In *Proceedings of ACM SIGCOMM '90*, pages 19–29, September 1990.

[12] M Sobel D. Heyman. *"Stochastic Models in Operations Research"*, volume Vol.I. McGraw Hill, 1982.

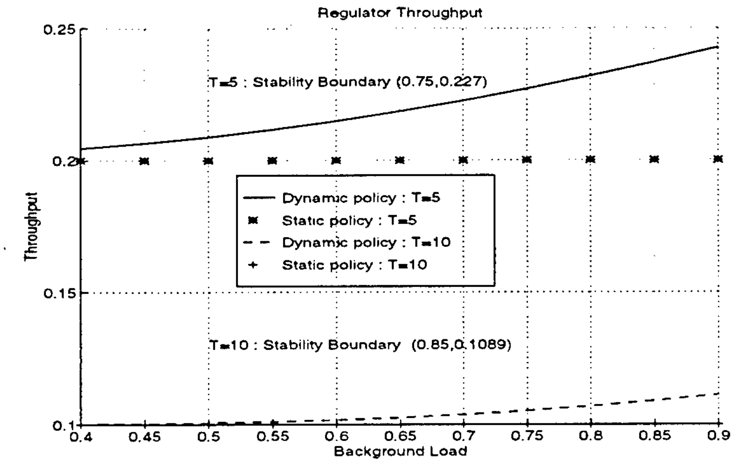


Figure 6: Regulator Throughput VS background utilization ρ_{back} ($\rho_{back} < 1$).

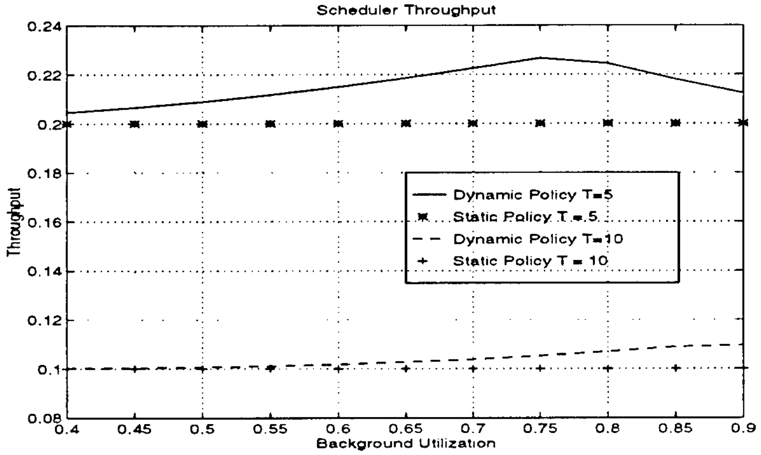


Figure 7: Scheduler Throughput VS utilization ρ ($\rho_{back} < 1$).

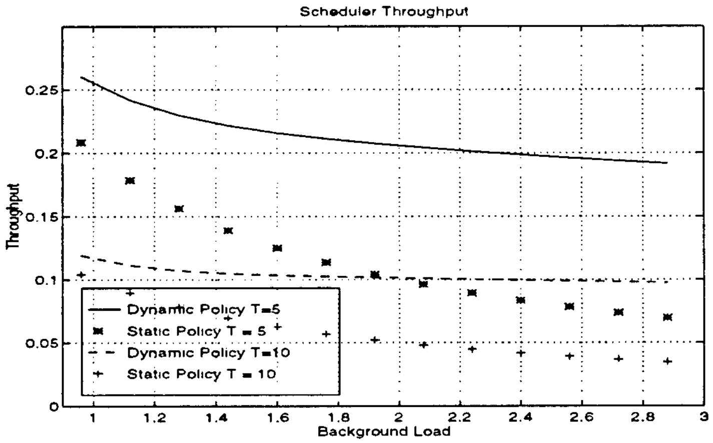


Figure 8: Scheduler Throughput VS background utilization ρ_{back}.

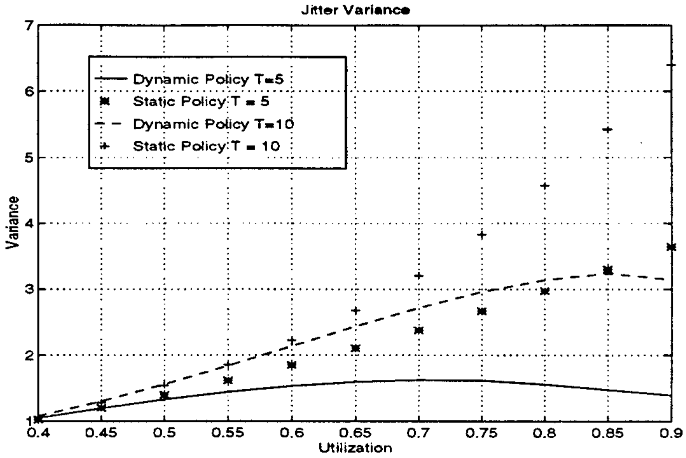


Figure 9: Variance VS background utilization ρ_{back} ($\rho_{back} < 1$).

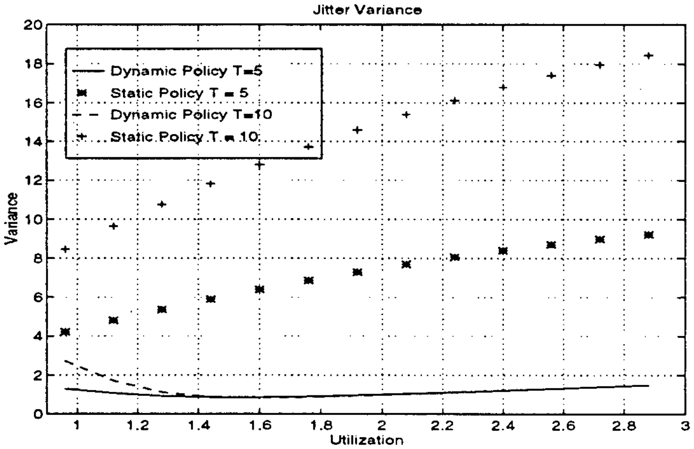


Figure 10: Variance VS background utilization ρ_{back} ($\rho_{back} > 1$). $P_b = 0.12 + (k-1) * 0.02$ where k the point of interest ($k = 1, 2, \ldots 13$)

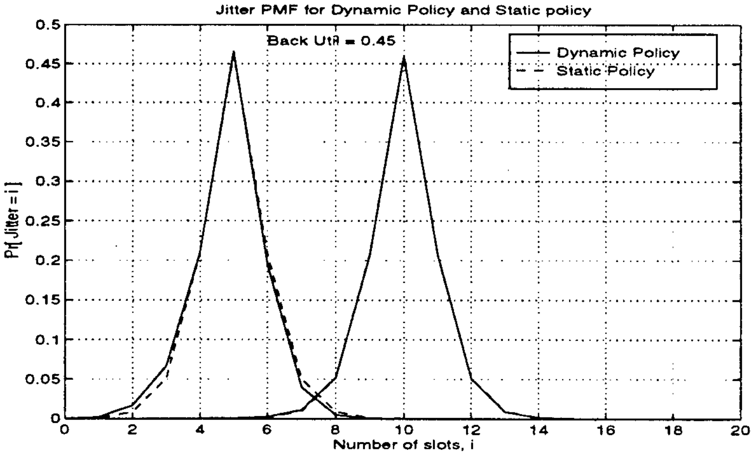


Figure 11: Low link utilization: Policies behave identically ($\rho_{back} < 1$)

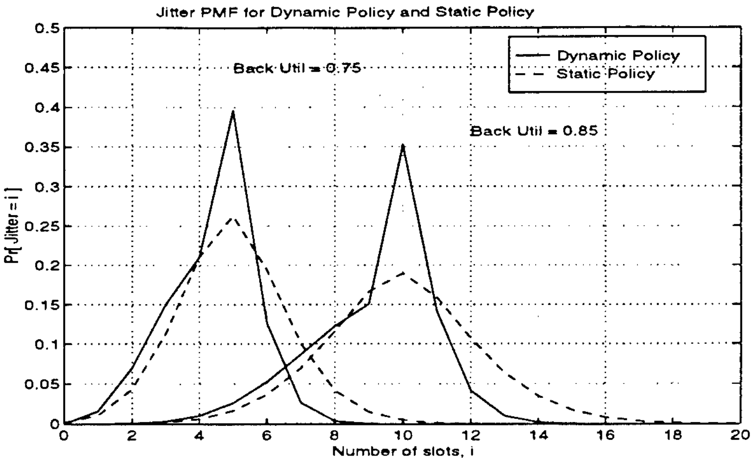


Figure 12: High link utilization: Jitter performance improves for the Dynamic Policy ($\rho_{back} < 1$)

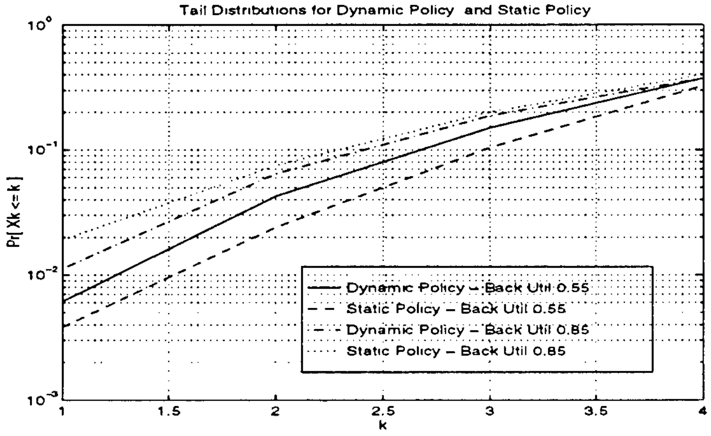


Figure 13: $Pr\{X_k \leq k\}$ for $k < T$, and $T = 5$ ($\rho_{back} < 1$)

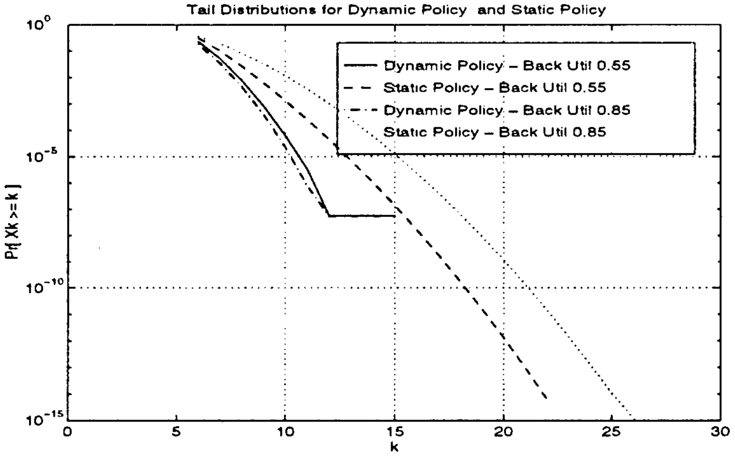


Figure 14: $Pr\{X_k \geq k\}$ for $k > T$, and $T = 5$ ($\rho_{back} < 1$)

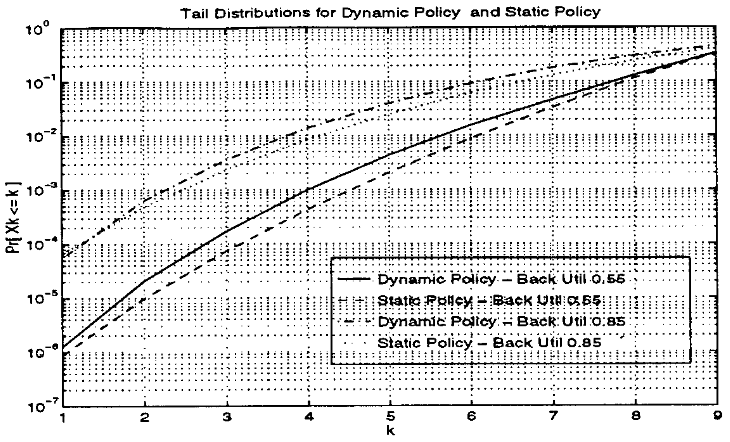


Figure 15: $Pr\{X_k \leq k\}$ for $k < T$, and $T = 10$ ($\rho_{back} < 1$)

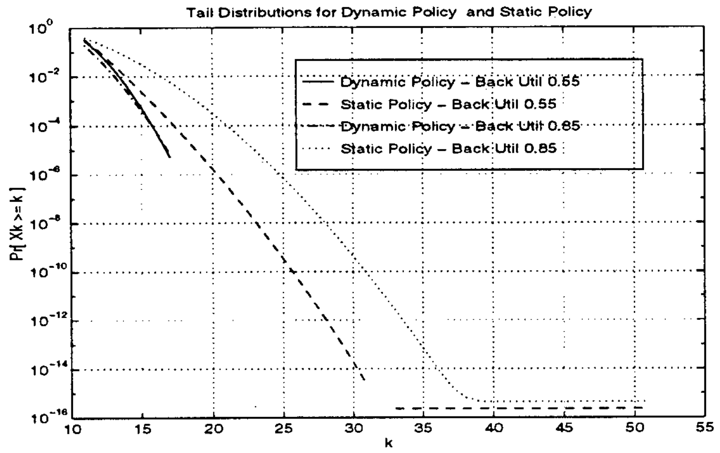


Figure 16: $Pr\{X_k \geq k\}$ for $k > T$, and $T = 10$ ($\rho_{back} < 1$)

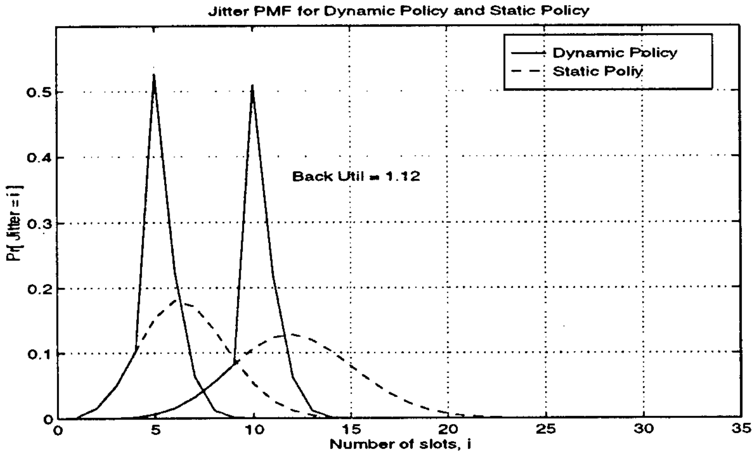


Figure 17: Jitter distributions under moderate overload ($\rho_{back} > 1$)

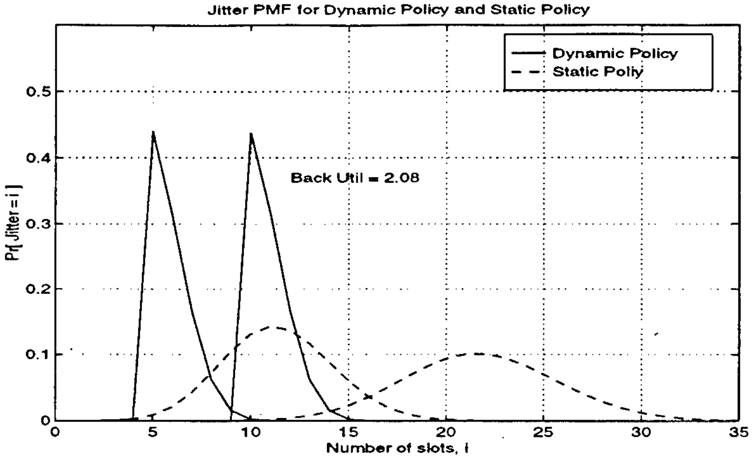


Figure 18: Jitter distributions under excessive overload ($\rho_{back} > 1$)

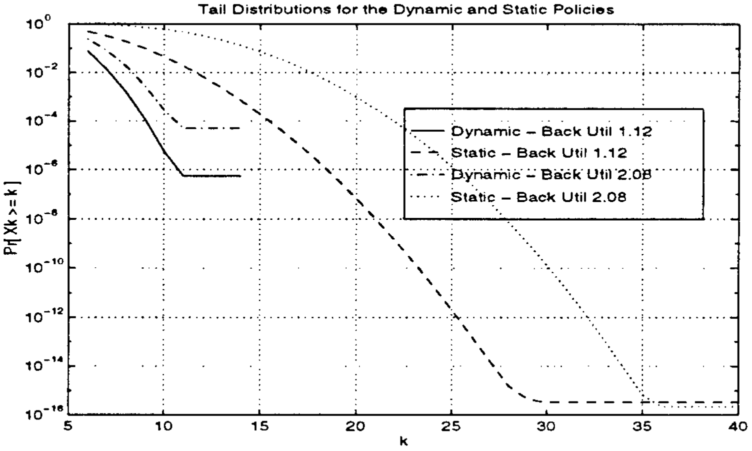


Figure 19: $Pr\{X_k \geq k\}$ for $k > T$, and $T = 5$ ($\rho_{back} > 1$)

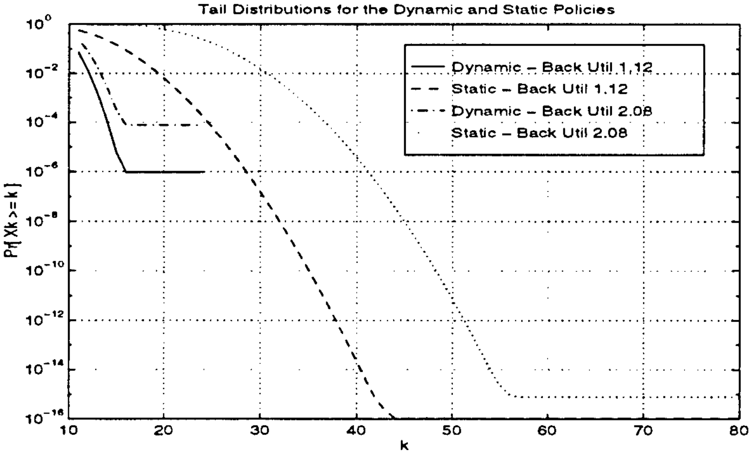


Figure 20: $Pr\{X_k \geq k\}$ for $k > T$, and $T = 10$ ($\rho_{back} > 1$)

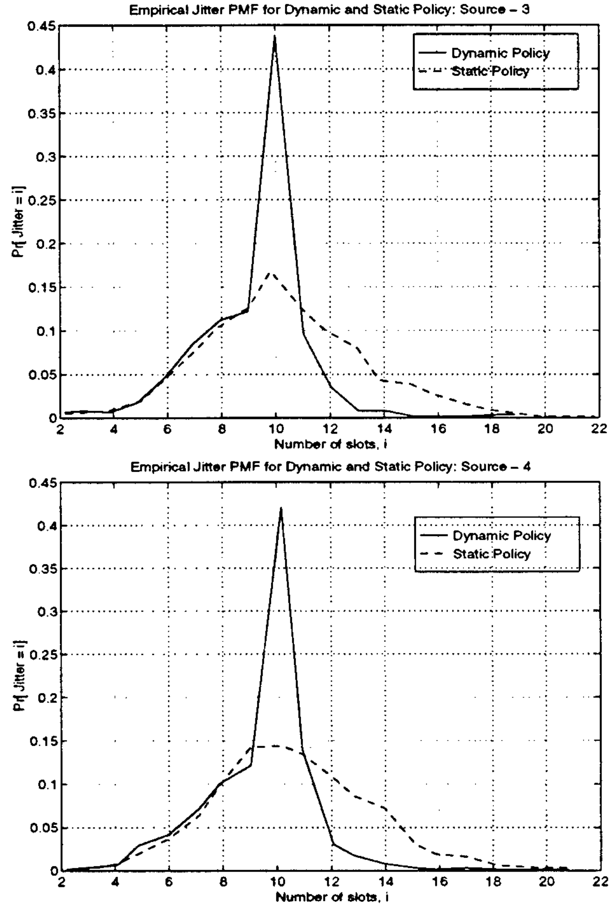
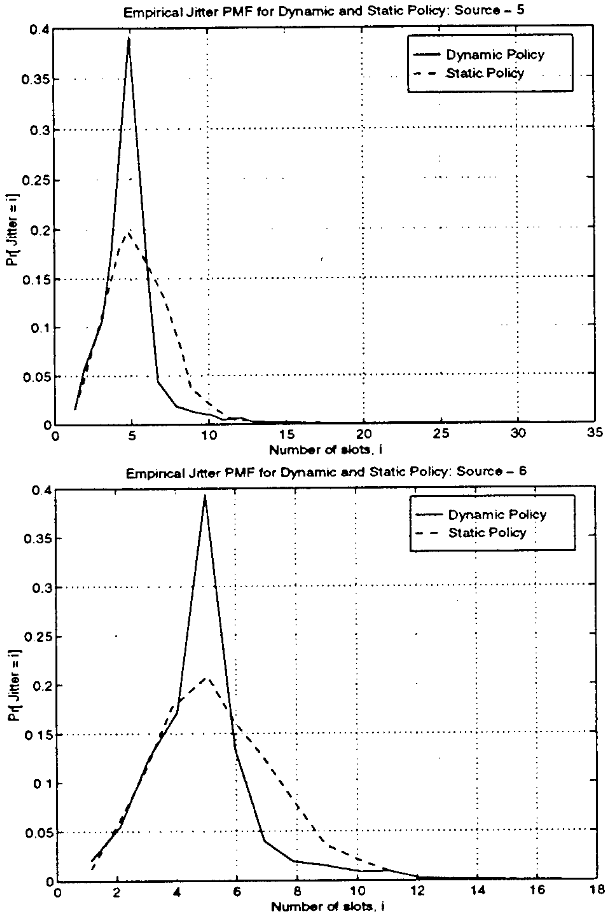


Figure 21: Empirical PMFs - 1^{st} simulation senarion


Empirical Jitter PMF for Dynamic and Static Policy: Source - 5

Empincal Jitter PMF for Dynamic and Static Policy: Source - 6

Figure 22: Empirical PMFs - 1^{st} simulation senarion

Study of Multiplexing for Group-based Quality of Service Delivery

Xin Wang and Ioannis Stavrakakis
Electrical & Computer Engineering,
Northeastern University,
360 Huntington Avenue,
Boston, MA 02115, U.S.A.
email: ioannis@cdsp.neu.edu

*

Abstract

In this work, the problem of scheduling packets with a group as opposed to an individual deadline is considered. Packets of the same group are supposed to belong in the same application data unit for which a deadline is set, and they are assumed to arrive over fixed time intervals (frames). The Frame-based Shortest Time to Extinction (F-STE) scheduling policy is considered at a multiplexer fed by multiple streams with group deadlines; expired packets are dropped (not served). The performance of the F-STE policy is evaluated in terms of the induced packet loss (deadline violation) probabilities. It is established that the F-STE policy outperforms the similarly simple First In First Out (FIFO) policy, as well as the potentially more complex FIFO policy which identifies and drops expired packets.

1 INTRODUCTION

High speed networks are today considered to be capable of supporting high-rate real-time applications such as video and multimedia. In order for the Quality of Service (QoS) of such applications to be delivered, networks must meet stringent performance requirements in terms of delay, delay jitter, throughput and packet loss. The timely delivery of packets associated with a real time application can be achieved in a number of ways. The direct approach would be to associate deadlines by which the packets must be transmitted and employ deadline-based

*Research supported in part by the Advanced Research Project Agency (ARPA) under Grant F49620-93-1-0564 monitored by the Air Force Office of Scientific Research (AFOSR).

scheduling mechanisms within the network. Examples of such scheduling policies are the Earliest Due Date (EDD) (Liu *et al.* 1973) (Lim *et al.* 1990) (Saito 1990), the Delay-EDD (Ferrari *et al.* 1990) and the Shortest-Time-to-Extinction (STE) (Panwar *et al.* 1988). Alternatively, scheduling policies which control the induced delay jitter or guarantee a certain transmission (service) rate could be employed. Such policies include the Virtual Clock (VC) (Zhang 1990) (Lam *et al.* 1995), Hierarchical Round Robin (HRR) (Kalmanek *et al.* 1990), Stop-and-Go (Golestani 1990), Rate Controlled Static Priority Queueing (RCSP) (Zhang *et al.* 1993), Fair Queueing (Demers *et al.* 1989), Weighted Fair Queueing (WFQ) (Takagi *et al.* 1991), also known as General Processor Sharing (GPS), and the packet based version of GPS, Packet-by-Packet Generalized Processor Sharing (PGPS) (Parekh *et al.* 1993) (Parekh *et al.* 1994). Since this paper proposes and investigates a new *deadline* driven scheduling policy, the related EDD and STE policies mentioned above are discussed in more detail.

The classical EDD policy minimizes the maximum lateness and maximum tardiness by transmitting packets in the order of their due dates. The lateness of a packet is defined to be the difference between the finishing time of its transmission and its due date. Its tardiness is given by $\max\{0, lateness\}$. The STE policy is very similar to the EDD policy in the sense that the packet with the earliest due date is served first. However, it differs from the EDD in that packets which miss their deadline (expired packets) are not transmitted. It has been established that the STE scheduling policy is optimal in the sense that it maximizes the fraction of packets transmitted before their deadline or, equivalently, minimizes the deadline violation probability.

In some applications, such as packetized voice or video, packets must be transmitted before their deadlines expire in order to be useful. On the other hand, these applications can tolerate the loss of a small fraction of packets. Clearly, the STE scheduling policy is optimal for such applications. However, implementation of the STE policy requires a time-consuming sorting process in order to identify the task with the earliest deadline. This makes the STE policy too complex for real-time realization in high speed networks. For this reason, the simple to implement FIFO policy is often employed. Although the performance of the FIFO scheduling policy is typically inferior to that of the STE policy with respect to the induced deadline violation probability, it is easy to establish that the STE policy and the FIFO policy that drops expired packets become identical if all packets sharing the same resource have identical time to extinction upon arrival. In such environments, the easy to implement FIFO which drops expired packets is also optimal in the sense that it minimizes the deadline violation probability.

The nature of most real-time applications suggests, however, that a deadline be associated with an application data unit (group of packets), such as a video-frame in video applications, rather than with individual packets. For example, in the VBR video encoder, the data for one horizontal strip of blocks is assembled into a single self-contained unit which is transmitted by means of multiple ATM cells (the unit is referred to as a video packet) (Verbiest *et al.* 1989). The address of the strip in the frame provides a time reference for decoder synchronization. For such applications, a deadline should be associated with each data unit rather than with each packet. Packets belonging in the same group now have variable times to extinction upon arrival, and the FIFO policy is non-optimal even if all the applications have identical QoS requirements.

In this work, the Shortest Time to Extinction (STE) scheduling policy is considered for scheduling packets with a group as opposed to individual deadline. Packets of the same group are defined here as packets arriving over a fixed time interval (frame). Packets belonging to the group with the earliest deadline (shortest time to extinction) are served first and the policy is called Frame-STE (F-STE) due to the central role of the underlying frame in determining the arrivals and the associated deadlines. It is easy to establish that the F-STE is equivalent to a dynamic Deadline-Ordered Head-of-Line (DO–HoL) priority scheduling policy. Clearly, F-STE is optimal in the sense that it minimizes the deadline violation probability. While a scheduler implementing the STE policy must search for the packet with the shortest time to extinction every time it serves a packet, no such search is necessary for the F-STE policy. This is an advantage intrinsic to group-based scheduling policies, in which the scheduling priority needs to be updated less often than in a packet-based scheduling policy, and has been pointed out by other authors (for example, (Lam *et al.* 1996)).

In an environment in which packets must be transmitted before their deadlines in order to be useful, a scheduling policy that drops expired packets is more effective than the one which serves expired packets, in the sense that a larger fraction of packets can be transmitted before their deadlines. In order for expired packets to be dropped in a FIFO scheduler, they typically need to be time-stamped and searched for, which adds significantly to the implementation complexity. In this paper, an implementation of the F-STE scheduler is outlined which does not require time-stamping and searching mechanisms. The performance of the F-STE policy is compared with that of the easy to implement (standard) FIFO policy which does not drop expired packets, as well as the more complex FIFO policy which drops expired packets.

Finally, it should be noted that the non-work-conserving version of the F-STE policy – considered also in this paper – resembles the Stop-and-Go scheduling policy (Golestani 1990). However, the objectives, the evaluated performance measures of interest and the work-conserving version of the F-STE policy considered in this paper are different from those in (Golestani 1990).

2 DESCRIPTION OF THE SYSTEM AND THE SCHEDULING POLICY

A network node with N incoming and one outgoing links is considered in this paper. All links are assumed to have the same capacity and thus a packet (fixed size information unit) transmission requires the same amount of time, referred to as the slot. Slot level system synchronization is assumed implying that packet arrivals and departures from the node occur at slot boundaries.

In addition to the time slot, a larger time constant of length T (in slots), called frame, is associated with each of the input streams. T takes integer values and is assumed to have the same value for all streams to simplify the analysis and the discussion. The relative shift of the frame boundaries of the incoming streams can be arbitrary. Without loss of generality and for simplicity, the frame boundaries are assumed to be uniformly distributed in this paper. Packet arrivals associated with the same stream and frame occur according to a Bernoulli process with a fixed rate. This rate may be different for different streams.

The frame is also assumed to modulate the QoS of the associated packets. Specifically, all packets of a stream arriving over the same frame are assumed to have the same deadline, set to be equal to the end of the next frame. Since the objective of this work is to determine whether significant performance gain can be achieved under the proposed policy, the deadline is set so that the analysis complexity be minimized. Packets which are not transmitted (served) by their deadline are dropped.

It is easy to see that packets with the same deadline (end of next frame) will have different times to extinction upon arrival, since their arrival times are different (one packet per slot per frame). When the frame boundaries associated with different streams are not synchronized, which would be typically the case, it is easy to establish that earlier arriving packets from one stream may have a larger time to extinction than later arriving packets from a different stream. As a result, the easy to implement FIFO policy would not serve packets in order of decreasing times to extinction and, for this reason, would not minimize the deadline violation probabilities; as said earlier, these probabilities are minimized under the STE policy.

In this paper, packets are served according to the STE policy. Because of the frame-structured packet arrivals and QoS definition, the STE policy – called in this environment Frame-based STE (F-STE) - is easy to implement, and thus, can be employed in a high-speed networking environment. Details of a simple implementation are presented in the next subsection.

The following definitions and alternative policy description will be useful in both the analysis and the implementation of the F-STE policy. Figure 1 depicts the arrival axes of $N = 4$ streams. The frames of all streams are identical (of length equal to T slots) and their boundaries are uniformly distributed. Let $\{t_k\}$ denote the sequence of frame boundaries from any streams, that is, t_k denotes the time when the *kth* frame ends. The deadline of packets associated with the frame ending at t_k is time t_{k+N}. Let I_k denote the stream whose frame ends at t_k; $I_k \epsilon \{0, 1, 2, \ldots, N-1\}$.

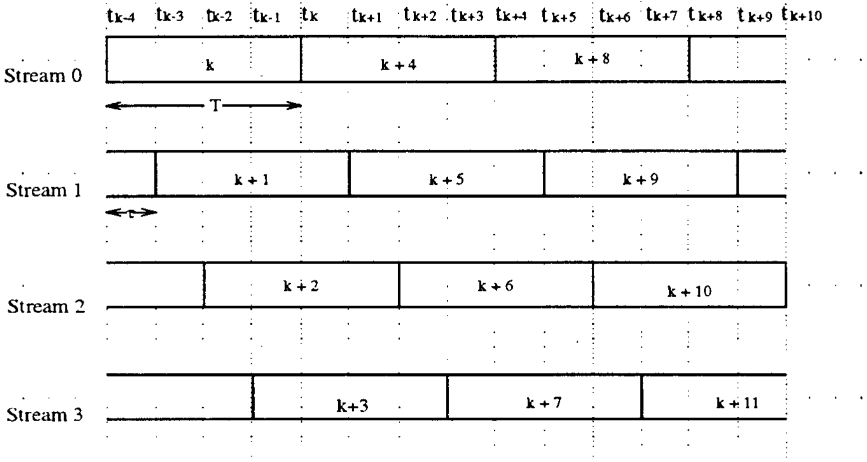


Figure 1 Frame structure of $N = 4$ traffic streams

According to the F-STE policy and in view of the above assumptions, the scheduler serves the streams according to the ordered HoL priority $(I_{k+1}, I_{k+2}, \ldots, I_{k+N})$ over the interval $(t_k, t_{k+1}]$. That is, high priority is given to packets from stream I_{k+1}, second highest priority is given to the packets from stream I_{k+2}, etc. It is easy to see that this policy serves packets in order of decreasing times to extinction. The ordered-HoL priority $(I_{k+1}, I_{k+2}, \ldots, I_{k+N})$ is cyclically shifted at the next frame boundary to $(I_{k+2}, I_{k+3}, \ldots, I_{k+N+1})$; and for this reason, the comprehensive service policy is called Dynamic-Ordered HoL priority policy (DO-HoL).

2.1 Implementation

The F-STE policy (or DO-HoL policy which drops expired packets) can be implemented without requiring a search for the packet with the shortest time to extinction, as described below and depicted in Figure 2.

Each stream is assigned a logically distinct data queue of capacity T which is the maximum number of packets delivered by the stream over one frame. At time t_k, the content of the data queue I_k is shifted to the service queue of capacity T. If the service queue overflows, the dropped packets will be precisely the ones which will miss their deadline. Packets accepted by the service queue will be served within T time units after their shifting since the service queue is served continuously.

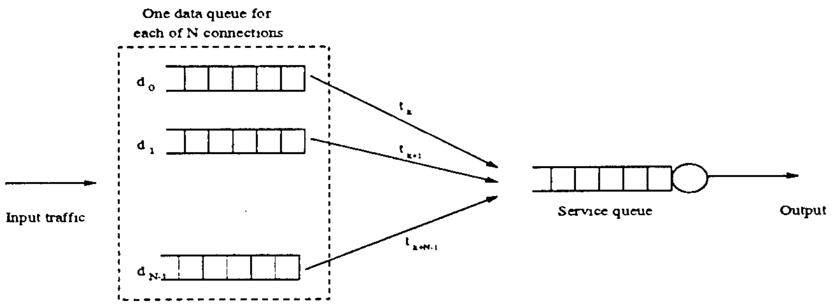


Figure 2 Implementation of the F-STE policy at a node with N incoming links (packets arriving over a frame accumulate in the corresponding data queue and are shifted to the service queue at the end of the frame).

If the service queue becomes empty at some time t, $t_k < t < t_{k+1}$, then the server either remains idle until data queue I_{k+1} is shifted into the service queue at t_{k+1} (non-work-conserving F-STE policy), or the server attends to the data queues according to the DO-HoL priority policy (work-conserving F-STE policy).

3 PERFORMANCE ANALYSIS

In this section, the packet loss (deadline violation) probability is derived for N streams served under the F-STE policy. The arrival process and QoS requirements are as described earlier.

Let τ_k denote the interval (in slots) between the two consecutive frame boundaries ending at t_k and t_{k+1}; that is $\tau_k = t_{k+1} - t_k$. In view of the assumptions in section 2, τ_k is independent of k and will be denoted by τ. Clearly, $\tau = \frac{T}{N}$. Let I_k denote the data queue whose content is to be shifted to the service queue at t_k. Let Q_k denote the service queue occupancy at time t_k, after the packets from data queue I_k are shifted into the service queue. The 2-dimensional process $\{Q_k, I_k\}_{k \geq 0}$ with state space $\{(i,j) : 0 \leq i \leq T, 0 \leq j \leq N-1\}$ is defined to be the *system* process and is employed in the analysis of the scheduling policies. Let $P(i,j,i',j')$ denote the transition probability that process $\{Q_k, I_k\}_{k \geq 0}$ moves from state (i,j) at time t_k to state (i',j') at time t_{k+1}. Note that this probability is zero if $j' \neq (j+1)\text{mod}(N)$, since only transitions from j to $j' = (j+1)\text{mod}(N)$ are possible. Throughout the paper, j' will always be equal to $(j+1)\text{mod}(N)$.

In the next subsection, the system process $\{Q_k, I_k\}_{k \geq 0}$ is determined under the non-work-conserving F-STE policy. In subsection 3.2, certain auxiliary processes are defined in order to obtain bounds or approximations on the evolution of the system process $\{Q_k, I_k\}_{k\geq 0}$ under the work-conserving F-STE policy. By employing the derivations of subsections 3.1 and 3.2, the packet loss (deadline violation) probability is derived for all cases in subsection 3.3. In the derivations that follow, A_m^k denotes the random variable representing the number of packets from stream k arrived over m slots. The probability mass function of A_m^k is given by the convolution of the Bernoulli random variables of the associated stream which are assumed to be of fixed value over a frame.

3.1 System equations for the non-work-conserving F-STE policy

Under this policy, packets can be served only after they have been shifted to the service queue, that is, only after the end of the frame over which they arrive. The server remains idle when the service queue is empty even if the data queues may be non-empty. This non-work-conserving version of the F-STE policy is considered primarily to facilitate the study of the F-STE policy presented afterwards. Nevertheless, this policy may be employed for applications in which the packets of the frame cannot be served before the packets of the entire frame have been received. It is easy to establish that the system process $\{Q_k, I_k\}_{k\geq 0}$ is a Markov chain, since I_k is a periodic Markov chain and Q_{k+1} is (probabilistically) determined from Q_k and A_m^k. The probability $P(i,j,i',j')$ that the Markov chain moves from state (i,j) to state (i',j') is given below. When $i \geq \tau$, the system queue cannot become empty between two successive shifts to the service queue, τ slots apart; when $i < \tau$, the service queue becomes empty between two successive shifts. Note that since only one packet can arrive over one slot, the maximum number of packets arriving over T time slots (one time frame) is T.

Case A : $\tau \leq i$

$$\text{For} \quad (i-\tau) \leq i' \leq T-1: \quad P(i,j,i',j') = P\{A_T^{j'} = i' - i + \tau\} \tag{1}$$

$$\text{For} \quad i' = T: \quad P(i,j,i',j') = \sum_{k=T-i+\tau}^{T} P\{A_T^{j'} = k\} \tag{2}$$

Case B : $0 \leq i < \tau$

$$\text{For} \quad 0 \leq i' \leq T: \quad P(i,j,i',j') = P\{A_T^{j'} = i'\} \tag{3}$$

3.2 Auxiliary system equations for the work-conserving F-STE policy

According to the work-conserving F-STE policy, if the service queue becomes empty at some time t, $t_k < t < t_{k+1}$, the server attends to the data queues according to the ordered HoL priority $(I_{k+1}, \ldots, I_{k+N})$ (see section 2). An implication of this policy is that the system process $\{Q_k, I_k\}_{k \geq 0}$ is not a Markov chain. Some of the arrivals to be shifted into the data queue at t_{k+1}, given by $A_T^{I_{k+1}}$, may have already been served while in the corresponding data queue. Since the latter quantity is needed to determine the evolution of $\{Q_k, I_k\}_{k \geq 0}$, $\{Q_k, I_k\}_{k \geq 0}$ is not a Markov chain.

In order to obtain the system evolution for the work-conserving case, the Markovian process $\{Q_k, I_k, \bar{Q}_k\}_{k \geq 0}$ can be considered, where $\bar{Q}_k$ is an $(N-1)$-dimensional vector representing the occupancy of the data queues which are not shifted to the service queue at time t_k; the one shifted is always 0 at t_k. The resulting analysis would be involved and the complexity would be prohibitive for a large N. Nevertheless, this approach has been employed in the study of the work-conserving F-STE policy supporting $N = 2$ streams. The derivation of the system equations for this case is presented in the appendix. Results for the two stream case are presented at the end of the paper.

The work-conserving F-STE policy for large N is studied here by following the approach described below. This approach leads to the derivation of bounds and approximations on the performance of the work-conserving F-STE policy with a complexity similar to that for the non-work-conserving case, even for large N.

Since all packet losses (deadline violations) are "registered" as service queue overflows occurring at some t_k (when the corresponding data queue is shifted into the service queue), bounds on the packet loss under the work-conserving F-STE policy can be obtained by defining proper auxiliary systems U and L which are such that their service occupancies at times t_k – denoted by Q_k^u and Q_k^l respectively – satisfy

$$Q_k^l \leq Q_k \leq Q_k^u. \tag{4}$$

For properly selected auxiliary systems (policies), the above relationship would then lead to a similar one involving the corresponding packet loss probabilities given by

$$L_k^l \leq L_k \leq L_k^u. \tag{5}$$

Furthermore, by constructing the auxiliary system in a way that the system processes posses the Markovian property, the packet loss probabilities will be easily derived, as described in subsection 3.3.

Since the non-work-conserving F-STE scheduler remains idle when the service queue becomes empty, and is otherwise identical to the work-conserving F-STE scheduler, it is evident that the associated system can serve as the auxiliary system U. In 3.2.1, an auxiliary system L is proposed, and equations are derived for the system evolution. Another auxiliary system A is proposed in 3.2.2, in order to obtain a close approximation for the packet loss of the work-conserving F-STE policy. The following will be satisfied by this auxiliary system.

$$Q^l_k \leq Q^a_k \leq Q^u_k, \quad L^l_k \leq L^a_k \leq L^u_k. \tag{6}$$

The tightness of the bounds and the accuracy of the approximate results are discussed in the last section where numerical results are presented.

Auxiliary system L

The auxiliary system L is defined to be similar to the system under the non-work-conserving F-STE policy with the following differences. If the service queue becomes empty before the next data queue shifting time t_{k+1}. the number of slots wasted under the non-work-conserving F-STE policy is considered as a service credit available at t_{k+1}; this amount is registered as an equivalent negative contribution to the service queue occupancy at t_{k+1}. There is an upper bound on the allowed credit accumulation equal to $T-\tau$. This bound is set by the requirement that earlier service credit cannot be utilized by future arrivals since it will be lost ("expired credit"). More specifically, service queue occupancy $x < \tau - T$ at t_k would imply the generation of at least $|x-(T-\tau)|$ credit units during the interval $(t_{k-N}, t_{k-(N-1)}]$; this credit must be wasted since all packets in the data queues at t_k have arrived after $t_{k-(N-1)}$.

The one-step transition probability for $\tau - T \leq i \leq T$ for system L is given by

For $\quad i' = T: \quad$ $$P(i,j,i',j') = \sum_{k=T-i+\tau}^{T} P\{A^{j'}_T = k\} \tag{7}$$

For $\quad (\tau - T) < i' < T-1: \quad$ $$P(i,j,i',j') = P\{A^{j'}_T = i' - i + \tau\} \tag{8}$$

For $\quad i' = \tau - T: \quad$ $$P(i,j,i',j') = \sum_{k=0}^{2\tau-i-T} P\{A^{j'}_T = k\} \tag{9}$$

Auxiliary system A

In deriving the evolution of system L, it was assumed that all the "unexpired" slots (inducing "unexpired" credit) wasted under the non-work-conserving policy could be utilized by serving the data queues. This is true only if at least one data queue is non-empty during the interval from when the service queue becomes empty, till the next shift to the service queue. In order to obtain a better approximation for the packet loss probability of the work-conserving F-STE policy, an auxiliary system A is considered, in which a more precise credit accumulation rule is applied. As in the case of the auxiliary system L, the service queue capacity i must satisfy $\tau - T \leq i \leq T$.

<u>Case A : $i' > 0$</u>

In this case, no service credit is generated.

$$\text{For} \quad (i-\tau) \leq i' \leq T-1 : \quad P(i,j,i',j') = P\{A_T^{j'} = i' - i + \tau\} \qquad (10)$$

$$\text{For} \quad i' = T : \quad P(i,j,i',j') = \sum_{k=T-i+\tau}^{T} P\{A_T^{j'} = k\} \qquad (11)$$

<u>Case B : $i' \leq 0$</u>

$[B1] : i > 0$

Unlike in system L, out of the $l = |i - \tau|$ non-used (by the packets in the service queue) service opportunities over $(t_k, t_{k+1}]$, only a portion will be registered as credit, based on the number of (earlier) packet arrivals to the data queues which have not been shifted to the service queue at, or before, t_k. Thus, the actual amount of credits kept will be equal to $\min(|l|, \tilde{A}_m)$ where $\tilde{A}_m = A^{j'+1}_{(N-2)\tau} + A^{j'+2}_{(N-3)\tau} + \ldots + A^{j'+N-2}_{\tau}$. Each of the terms in $\tilde{A}_m$ represents packet arrivals over some τ and for some stream which have not been shifted to the service queue at, or before, t_k. Arrivals over the interval $(t_k, t_{k+1}]$ over which this credit is generated are not considered since some of them may not arrive before the credit is generated; clearly, this is an approximation. In this case, the only possible transitions are to state i', where $-\tau < i' \leq 0$, and their probabilities are given by

$$P(i,j,i',j') = P\{A_T^{j'} = i' - i + \tau\} P\{\tilde{A}_m \geq |i'|\} + \sum_{k=0}^{i'-i+\tau-1} P\{A_T^{j'} = k\} P\{\tilde{A}_m = |i'|\} \qquad (12)$$

$[B2] : \tau - T \leq i \leq 0$

The service queue is empty at t_k, and $|i|$ service credits have already been accumulated. Let $l = i' - i$. If $l < 0$, $|l|$ is equal to the credit generated over $(t_k, t_{k+1}]$. Since packet arrivals to the data queues occuring before t_{k-1} have been considered toward the determination of the original credit $|i|$ (as indicated in the previous case, $[B1]$), only packet arrivals to the data queues occuring over $(t_{k-1}, t_k]$ will be used to determine the actual new credit to be kept. The latter arrivals are given by $\tilde{A_n} = A_T^{j'+1} + A_T^{j'+2} + \ldots + A_T^{j'+N-2}$. Finally, no credit is generated if $l \geq 0$. The transition probabilities are given by:

For $\tau - T < i' \leq 0$:

$$P(i,j,i',j') = \begin{cases} P\{A_T^{j'} = \tau + l\} & l > 0 \\ P\{A_T^{j'} = \tau + l\}P\{\tilde{A_n} \geq |l|\} + & \\ \sum_{k=0}^{\tau+l-1} P\{A_T^{j'} = k\}P\{\tilde{A_n} = |l|\} & l \leq 0 \end{cases} \tag{13}$$

For $i' = \tau - T$:

$$P(i,j,i',j') = \begin{cases} P\{A_T^{j'} = \tau + l\} & l > 0 \\ \sum_{k=0}^{\tau+l} P\{A_T^{j'} = k\}P\{\tilde{A_n} \geq |l|\} & l \leq 0 \end{cases} \tag{14}$$

3.3 Computation of packet loss probabilities

In view of the Markovian structure for the system processes defined in the previous subsections, the induced packet loss (deadline violation) probabilities are easy to determine. As indicated earlier, the packets whose deadline expires are precisely the packets which overflow from the service queue upon shifting to that queue.

Let L_{ij} denote the number of packets dropped over the interval $(t_k, t_{k+1}]$, where (i,j) is the system state at t_k. Notice that all losses will be associated with packets from stream $j' = (j+1)\text{mod}(N)$. Clearly,

$$L_{ij} = \left(A_T^{j'} + (i - \tau) - T\right)^+ . \tag{15}$$

Notice that at most $i - \tau$ packets may be lost since the maximum number of per stream and per frame arrivals is equal to T; no loss occurs if $i \leq \tau$. The average value of L_{ij}, $\bar{L_{ij}}$, is given by

$$\bar{L_{ij}} = \sum_{k=1}^{i-\tau} kP(A_T^{j'} = T - (i - \tau) + k). \tag{16}$$

The loss rate for stream $j' = (j+1)\text{mod}(N)$ over $(t_k, t_{k+1}]$, when the system state at t_k is (i,j), is given by

$$R_{ij} = \frac{\bar{L_{ij}}}{\lambda^{j'}T}. \tag{17}$$

where $\lambda^{j'}$ denotes the arrival rate of stream j' per slot. Finally, the packet loss probability is given by

$$L = \sum_{i=\tau+1}^{T} \sum_{j=0}^{N-1} R_{ij}\pi(i,j), \tag{18}$$

where $\pi(i,j)$ is the steady state probability distribution of the corresponding Markov chain.

4 NUMERICAL RESULTS

In this section, two sets of numerical results are presented. Figures 3 to 6 are derived to evaluate the accuracy of the bounds and approximations associated with the analysis of the general F-STE policy, as well as to present exact results for the non-work-conserving F-STE policy, and the work conserving F-STE policy with $N = 2$. Figures 7 and 8 are derived for the comparative evaluation of the F-STE and the FIFO policies.

The exact packet loss probability vs system utilization λ (for $T = 8$) for $N = 2$ streams served under the work-conserving F-STE are shown in Figure 3; simulations are also shown for verification of the equation. Results for systems U (non-work-conserving F-STE policy) and L are also shown.

Results for the packet loss probability for $N = 4$ streams vs system utilization λ (for $T = 8$) and vs frame length (for $\lambda = 0.88$) are presented in Figures 4 and 5, respectively. Results are shown for systems L, U, A, and simulations. For the cases considered here, the approximate system (system A) seems to provide for an accurate approximation of the performance of the work-conserving F-STE policy. The lower bound (auxiliary system L) is observed to be tighter than the upper bound (auxiliary system U, that is the non-work-conserving policy). As λ increases, the tightness of the lower bound improves since more of the unexpired credit in system L will actually be used under the work-conserving F-STE policy. Similarly, the tightness of the upper bound (non-work-conserving system) improves as λ increases since the server will be found idle less often and, thus, the work-conserving and non-work-conserving policies will tend to become identical. As T increases, the tightness of the upper bound deteriorates since the work-conserving and non-work-conserving F-STE policies become increasingly different and the server is found more frequently idle under system U while there is work in the system. The slight improvement of the tightness of the lower bound as T increases may be attributed to an increasing probability that a data queue is non empty and thus increasing amount of credit is kept. The impact of the number of streams N on the induced packet loss probability is illustrated in Figure 6 vs frame length T (for

$\lambda = 0.88$); the results are for the work-conserving policy ($N = 2$, exact analysis) and for the approximate system A ($N = 4, 8, 16$).

Simulation results for the induced packet loss probability under a simple FIFO policy, the FIFO policy in which expired packets are dropped, and the work-conserving F-STE policy are shown in figures 7 and 8 vs system utilization λ (for $T = 16$ and $N = 4$) and vs frame length T (for $\lambda = 0.88$ and $N = 4$), respectively. The results from the analysis for the approximate system A are also shown. All results are derived for symmetric load.

As expected, the packet loss probability of all three scheduling policies decreases as the system utilization decreases. The packet loss probability also decreases as the frame length increases. The latter may be attributed to the associated increase -as the frame length increases- in the time to extinction as well as the increased smoothness of the cumulative traffic over a frame.

The FIFO policy which drops expired packets is seen to induce a smaller packet loss probability than the standard FIFO policy, as expected. The advantage is more pronounced when the scheduler is operating nearly at maximum capacity where the system utilization is high and the time to extinction is short (that is, the frame length is short).

As expected, the work-conserving F-STE policy outperforms both FIFO policies. It should be noted that the F-STE scheduler is comparable in terms of implementation complexity to the scheduler that implements the standard FIFO policy (which does not drop expired packets). Comparison of these two policies (Figures 7 and 8) shows that the F-STE policy always outperforms the FIFO policy by a large margin, irrespective of frame length and system utilization.

Comparison of the F-STE policy with the more complex FIFO policy which drops expired packets reveals that for high system utilization (Figure 7) the performance difference between F-STE and FIFO is small due to the scheduler throughput limitation. As λ decreases, the sub-optimality of the FIFO becomes a factor with increasing weight in inducing losses. The difference in performance between the above two policies is seen to increase as the frame length increases (Figure 8). This may again be attributed to increased sub-optimality of the FIFO policy: as the frame length increases, the range of extinction times of packets of the same frame increases leading to increased sub-optimality of the FIFO policy.

Evidently, the performance difference between the two scheduling policies that drop expired packets decreases under conditions (small frame length, high system utilization) in which the packet loss is largely due to the scheduler limitation, rather than the sub-optimality of FIFO. The performance of the standard FIFO policy remains significantly worse even under these conditions; this may be attributed to the additional sub-optimality of this policy resulting from transmitting packets that have exceeded their deadlines which becomes significant under these conditions.

APPENDIX

A System evolution for 2 stream case (exact analysis)

The accurate calculation of the packet loss probability for the F-STE policy in the case of a node with two streams is described in this appendix. As mentioned in subsection 3.2, the Markov process $\{Q_k, I_k, \bar{Q}_k\}_{k \geq 0}$ (denoted as process M_1) embedded at $\{t_k\}_{k \geq 0}$ needs to be considered. Q_k denotes the service queue occupancy after packets have been shifted in from data queue I_k at t_k. $\bar{Q}_k$ is an $(N-1)$-dimensional vector representing the occupancy of the data queues at time t_k which have not been shifted to the service queue; that is, $\bar{Q}_k = (Q_k^{I_{k+1}}, Q_k^{I_{k+2}}, \ldots, Q_k^{I_{k+N-1}})$, where Q_k^j denotes the occupancy of data queue j at t_k. Notice that $Q_k^{I_k} = 0$ since this data queue is being emptied (shifted to the service queue) at t_k and thus there is no need to include it in the $(N-1)$-dimensional process. The state space of M_1 for $N = 2$ is given by $S_2 = \{(i,j,l) : 0 \leq i \leq T, 0 \leq j \leq N-1, 0 \leq l \leq \tau\}$. Let $P(i,j,l;i',j',l')$ denote the probability that Markov chain M_1 moves from state (i,j,l) at t_k to state (i',j',l') at t_{k+1}, (i,j,l), $(i',j',l') \epsilon S_2$. These probabilities are described below where different cases are considered based on the starting state of the service queue $(Q_k = i)$.

<u>Case A : $\tau \leq i$</u> (service queue will not become empty before t_{k+1})

For $(i-\tau) \leq i' \leq T-1$:

$$P(i,j,l;i',j',l') = P\{A_T^{j'} = i' - i + \tau\}P\{A_\tau^j = l'\} \tag{19}$$

For $i' = T$:

$$P(i,j,l;i',j',l') = \sum_{k=T-i+\tau}^{T} P\{A_T^{j'} = k\}P\{A_\tau^j = l'\} \tag{20}$$

<u>Case B : $0 \leq i < \tau$</u> (service queue will become empty before t_{k+1})

In this case, the service queue becomes empty at time $t_k + i$, and remains empty (for a period of $\tau - i$ slots) till the next shift to the service queue at t_{k+1}. Over the interval $(t_k + i, t_{k+1}]$, packets are served from the data queues according to the HoL priority $(I_{k+1}, I_{k+2}) = (j', j)$. Two cases need to be considered depending on whether $i' = 0$ or $i' > 0$ as explained below.

Case B.1 : $0 \leq i < \tau$, $i' > 0$

In this case, data queue $I_{k+1} = j'$ is always nonempty over $(t_k + i, t_{k+1}]$ since otherwise $i' = 0$. The latter is seen by noting that data queue j' will remain empty after it becomes empty for the first time within the interval $(t_k + i, t_{k+1}]$ – and, thus, $i' = 0$ – since stream j' generates at most one packet per slot and receives HoL priority over the interval $(t_k + i, t_{k+1}]$. As a result, no other data queue receives any service over this interval when $i' > 0$ and the transition probabilities of M_1 depend only on the cumulative arrivals over T for stream j' and τ for stream j, and they are identical to those under Case A above.

Case B.2 : $0 \leq i < \tau$, $i' = 0$

In this case, data queue $I_{k+1} = j'$ becomes empty at some slot in $(t_k + i, t_{k+1}]$ (and remains empty thereafter). During the slots in which data queue $I_{k+1} = j'$ is empty *and* no packet is generated by stream j' (which would be served under the HoL priority), the other queue will be served if nonempty. In order to determine the content of data queue j at t_{k+1}, to determine the state of $\bar{Q}_{k+1} = Q^j_{k+1}$ (since $I_{k+1} = j'$ and $j = (j' + 1) \bmod (2)$), the slot by slot evolution of data queue j needs to be followed over the interval $(t_k + i, t_{k+1}]$. Since the evolution of this queue depends on whether data queue j' is empty or not, the evolution of both queues over $(t_k + i, t_{k+1}]$ is considered by using the auxiliary Markov chain defined next.

Let $(\hat{Q}^{j'}_0, \hat{Q}^j_0)$ denote the occupancies of the corresponding data queues when the service queue becomes empty (at $t_k + i$). Let $\{\hat{Q}^{j'}_n, \hat{Q}^j_n\}_{n \geq 0}$ denote the data queue occupancy process embedded at the *slot* boundaries with initial state $(\hat{Q}^{j'}_0, \hat{Q}^j_0)$. This process (referred to as process M_2) evolves as a Markov chain over the interval $(t_k + i, t_{k+1}]$ and its state after $\tau - i$ transitions, $(\hat{Q}^{j'}_{\tau-i}, \hat{Q}^j_{\tau-i})$, is such that : $(Q_{k+1}, I_{k+1}, \bar{Q}_{k+1}) = (\hat{Q}^{j'}_{\tau-i}, j', \hat{Q}^j_{\tau-i})$. That is, the $(\tau - i)$-step transition of M_2 will determine the state of M_1 at t_{k+1}. The 1-step and m-step transition probabilities of M_2 are derived below. Let the state space of M_2 be defined by $\hat{S}_2 = \{(\hat{q}^{j'}_n, \hat{q}^j_n) : 0 \leq \hat{q}^{j'}_n \leq T, 0 \leq \hat{q}^j_n \leq \tau\}$. Let A^j denote the Bernoulli random variable describing the number of packets generated over a slot by stream j.

For $0 < \hat{Q}^{j'}_n \leq T$, data queue j' will be served over the current slot and thus the evolution of the queues is given by

$$\hat{Q}^{j'}_{n+1} = \hat{Q}^{j'}_n - 1 + A^{j'}, \tag{21}$$

$$\hat{Q}^j_{n+1} = \hat{Q}^j_n + A^j. \tag{22}$$

Thus, the 1-step transition probability of M_2 is given by

$$P_{M_2}(\hat{q}_n^{j'}, \hat{q}_n^j; \hat{q}_{n+1}^{j'}, \hat{q}_{n+1}^j) = P\{A^{j'} = \hat{q}_{n+1}^{j'} - \hat{q}_n^{j'} + 1\} \\ P\{A^j = \hat{q}_{n+1}^j - \hat{q}_n^j\} \tag{23}$$

for $0 < \hat{q}_n^{j'} \leq T$, $0 \leq \hat{q}_{n+1}^{j'} \leq T$, $0 \leq \hat{q}_n^j, \hat{q}_{n+1}^j \leq \tau$.

For $\hat{Q}_n^{j'} = 0$, data queue j will be served if nonempty *and* no packet is generated by stream j' over the current slot. Thus the evolution of the queues is given by

$$\hat{Q}_{n+1}^{j'} = 0, \tag{24}$$

$$\hat{Q}_{n+1}^j = (\hat{Q}_n^j - 1_{\{A^{j'}=0\}})^+ + A^j. \tag{25}$$

Thus, the 1-step transition probability of M_2 is given by

$$P_{M_2}(0, \hat{q}_n^j, 0, \hat{q}_{n+1}^j) = P\{A^{j'} = 1\}P\{A^j = \hat{q}_{n+1}^j - \hat{q}_n^j\} + \\ P\{A^{j'} = 0\}P\{A^j = \hat{q}_{n+1}^j - \hat{q}_n^j + 1\} \tag{26}$$

for $0 < \hat{q}_n^j \leq \tau$, $0 \leq \hat{q}_{n+1}^j \leq \tau$, and

$$P_{M_2}(0, 0; 0, \hat{q}_{n+1}^j) = P\{A^{j'} = 1\}P\{A^j = \hat{q}_{n+1}^j\} + \\ P\{A^{j'} = 0\}1_{\{\hat{q}_{n+1}^j=0\}} \tag{27}$$

for $0 \leq \hat{q}_{n+1}^j \leq 1$.

The m-step transition probability of M_2 can be derived recursively in terms of the $(m-1)$-step and the 1-step transition probabilities and it is given by

$$P_{M_2}^m(\hat{q}_n^{j'}, \hat{q}_n^j; \hat{q}_{n+m}^{j'}, \hat{q}_{n+m}^j) = \\ \sum_{\hat{q}_{n+m-1}^{j'}=0}^{T} \sum_{\hat{q}_{n+m-1}^j=0}^{\tau} P_{M_2}^{m-1}(\hat{q}_n^{j'}, \hat{q}_n^j; \hat{q}_{n+m-1}^{j'}, \hat{q}_{n+m-1}^j) \\ P_{M_2}(\hat{q}_{n+m-1}^{j'}, \hat{q}_{n+m-1}^j; \hat{q}_{n+m}^{j'}, \hat{q}_{n+m}^j) \tag{28}$$

for $0 \leq \hat{q}_n^{j'}, \hat{q}_{n+m-1}^{j'}, \hat{q}_{n+m}^{j'} \leq T$, $0 \leq \hat{q}_n^j, \hat{q}_{n+m-1}^j, \hat{q}_{n+m}^j \leq \tau$.

Finally, the transition probabilities of process M_1 for case B.2 are given by

$$P(i,j,l;0,j',l') = \sum_{k1=0}^{i}\sum_{k2=0}^{i} P\{A_i^{j'} = k_1\}P\{A_i^{j} = k_2\} \\ P_{M_2}^{\tau-i}(l+k_1,k_2;0,l'), \qquad (29)$$

where A_i^j denotes the number of packet arrivals from stream j over i slots with $A_0^j \triangleq 0$.

B Computation of packet loss probability

The loss probability can be calculated by following a similar approach as in subsection 3.3. Let (i,j,l) be the system state at time t_k. The number of packet losses from stream $j' = (j+1)\text{mod}(2)$ over the interval $(t_k, t_{k+1}]$ is given by

$$L_{ijl} = (A_T^{j'} + (i-\tau) - T)^{+}. \qquad (30)$$

Notice that only packets from stream j' may expire over this interval. The average value of L_{ijl} is given by

$$\bar{L}_{ijl} = \sum_{k=1}^{i-\tau} kP(A_T^{j'} = T-(i-\tau)+k). \qquad (31)$$

The loss rate for stream j' over $(t_k, t_{k+1}]$, when the system state at t_k is (i,j,l), can now be written as

$$R_{ijl} = \frac{\bar{L}_{ijl}}{\lambda^{j'}T}. \qquad (32)$$

The total loss probability is therefore

$$L = \sum_{i=\tau+1}^{T}\sum_{j=0}^{N-1}\sum_{l=0}^{\tau} R_{ijl}\pi(i,j,l), \qquad (33)$$

where $\pi(i,j,l)$ is the steady state probability distribution of the Markov chain M_1.

REFERENCES

L. Zhang (1990) "Virtual Clock: A New Traffic Control Algorithm for Packet. Switching Networks", *Proceedings of SIGCOMM'90*, pp. 19-29, 1990.

Simon S. Lam and Geoffrey G. Xie (1995) "Burst Scheduling : Architecture and Algorithm for switching packet video", *Technical Report TR-94-20, Department of Computer Sciences, UT-Austin*, Revised, January 6 1995.

Simon S. Lam and Geoffrey G. Xie (1996) "Group Priority Scheduling", *Proceedings of IEEE INFOCOM'96, San Francisco*, pp. 1346-1356, June 1996.

W. Verbiest, L. Pinneboo (1989) "A Variable Bit Rate Video Codec for Asynchronous Transfer Mode Networks", *IEEE Journal on Selected Areas in Communications*, pp. 761-770, June 1989.

C. R. Kalmanek, H. Kanakia, S. Keshav (1990) "Rate Controlled Servers for Very High-Speed Networks", *Proceedings of IEEE GLOBECOM'90*, pp. 300.3.1 - 300.3.9 1990.

S. Jamaloddin Golestani (1990) "A Stop-and-Go queueing framework for congestion management", *Proceedings of IEEE INFOCOM'90, San Francisco*, pp. 527-542, June 1990.

D. Ferrari, P. Verma (1990) "A Scheme for Real-Time Channel Establishment in Wide-Area networks", *IEEE Journal on Selected Areas in Communications*, pp. 368-379, April 1990.

C. L. Liu, J. W. Layland (1973) "Scheduling Algorithms for Multiprogramming in a Hard Real Time Environment", *Journal of the ACM*, pp. 46-61, Jan., 1973.

Y. Lim, J. Kobza (1990) "Analysis of a Delay-Dependent Priority Discipline in a Integrated MultiClass Traffic Fast Packet Switch", *IEEE Transactions on Communications*, pp. 659-665, May 1990.

H. Saito (1990) "Optimal Queueing Discipline for Real-Time Traffic at ATM Switching Nodes", *IEEE Transactions on Communications*, pp. 2131-2136, Dec. 1990.

S. S. Panwar, D. Towsley, J. K. Wolf (1988) "Optimal Scheduling Policies for a Class of Queues with Customer Deadlines to the Beginning of Service", *Journal of the ACM*, pp. 832-844, 1988.

A. Demers, S. Keshav, S. Shenker (1989) "Analysis and Simulation of Fair Queuing Algorithm", *Proceedings of SIGCOMM*, pp. 1-12, 1989.

A. K. Parekh, R. G. Gallager (1993) "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: the Single Node Case", *IEEE/ACM Transactions on Networking*, pp. 344-357, June 1993.

A. K. Parekh, R. G. Gallager (1994) "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: the Multiple Node Case", *IEEE/ACM Transactions on Networking*, pp. 137-150, April 1994.

Y. Takagi, S. Hino, T. Takahashi (1991) "Priority Assignment Control of ATM Line Buffers with Multiple QoS Classes", *IEEE Journal on*

Selected Areas in Communications, pp. 1078-1092, Sep. 1991.

H. Zhang, D. Ferrari (1993) "Rate Controlled Static Priority Queueuing", *Proceedings of IEEE INFOCOM'93, San Francisco*, pp. 227-236, March 1993.

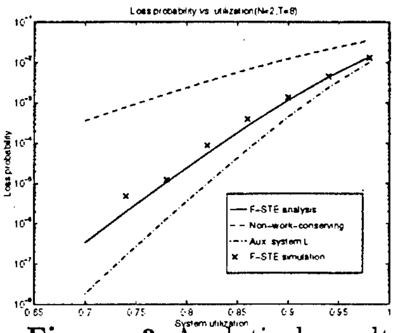


Figure 3 Analytical results for loss probability vs system utilization for $N = 2$ and $T = 8$.

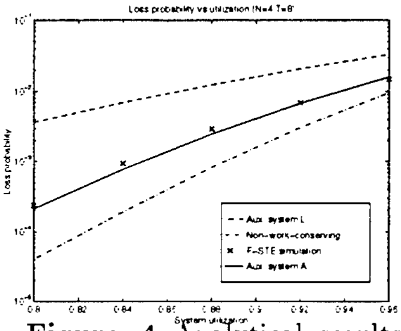


Figure 4 Analytical results for loss probability vs system utilization $N = 4$ and $T = 8$.

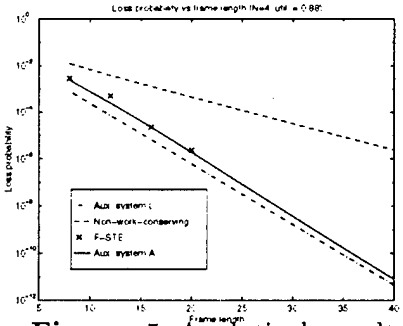


Figure 5 Analytical results fc. loss probability vs frame length $N = 4$ and system utilization equal to 0.88.

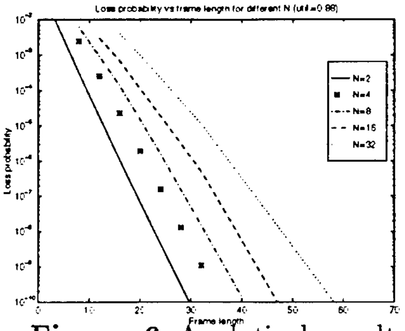


Figure 6 Analytical results for loss probability vs frame length, for different values of N given by system A for system utilization equal to 0.88.

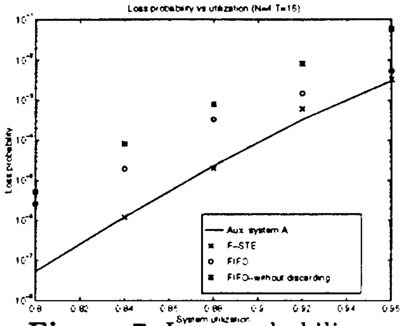


Figure 7 Loss probability vs system utilization for F-STE and FIFO for $N = 4$ and $T = 16$.

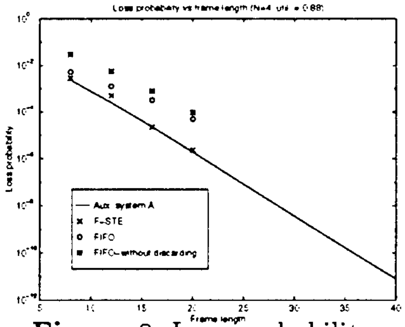


Figure 8 Loss probability versus frame length for F-STE and FIFO for $N = 4$ and system utilization equal to 0.88.

On the Impact of Time Scales on Feedback Based Rate Control *

Tamer Dağ † Ioannis Stavrakakis †

†Dept. of Electrical and Computer Engineering
Northeastern University
Boston, MA 02115
e-mail : {ioannis,tdag}@cdsp.neu.edu

Abstract

Since delay tolerable, the ABR applications can be allocated the remaining resources after CBR (Constant Bit Rate) and VBR (Variable Bit Rate) applications have been accommodated. To avoid excessive losses, the transmission rates of the ABR applications should be modulated by the amount of remaining resources. That is, the ABR rates should be controlled through a feedback based rate control mechanism. In this paper, a network link shared by remote ABR and VBR applications is considered and the impact of various system time scales on the effectiveness of the feedback based flow control scheme is investigated. These time scales are expressed in terms of the network transmission speed, the minimum tolerable ABR rate and the rate of change of the VBR source rate. While the negative impact of a decreased network time scale on the effectiveness of this control scheme is well known, the impact of the ABR and VBR time scales has not been investigated in the past. It turns out that for a given network time scale, the induced cell losses can be significantly reduced for increased ABR and/or VBR time scales and thus, the latter time scales should be taken into consideration when evaluating the effectiveness of an adaptive feedback based rate control mechanism.

1 Introduction

As indicated by the ATM Forum, the ABR service class allows non-real time, delay tolerant traffic sources to share the available bandwidth not utilized by the VBR and CBR service classes. Due to the fluctuations in the bandwidth usage of VBR/CBR sources, the ABR source rates should be controlled through a feedback based rate control mechanism.

When the ABR source receives the feedback generated by the receiving end or a network node, it adjusts its rate to the current available bandwidth according to the traffic control policy in effect [1], [2], [3]. Several traffic control policies have been proposed for the rate based flow control such as the Explicit Forward Congestion Indication (EFCI) [4], Backward Explicit Congestion Notification (BECN) [5], Proportional Rate Control Algorithm (PRCA) [6], Explicit Rate Feedback (ERF) [7] and Enhanced Proportional Rate Control Algorithm (EPRCA) [8].

Under the feedback based rate control policy used in this paper, a feedback carrying information about the available bandwidth is transmitted to the ABR source, every time that the available bandwidth for the ABR source changes. If the propagation delay between the ABR source and the network access node is non-negligible, then the ABR source will transmit at a rate based on the most recent level of available bandwidth until it receives the feedback. For example, if the propagation delay between the ABR source and the network access node is T_d and the available bandwidth changes at time t, the ABR source will learn about the change in bandwidth availability at time $t + T_d$. The adjusted rate from the ABR source will reach the network access node at time $t + 2T_d$, assuming that the backward and forward propagation delays are equal. Therefore, for a roundtrip propagation delay of $2T_d$ a bandwidth mismatch will occur. During a bandwidth mismatch the capacity of the link will either be exceeded (overutilization) or will be used inefficiently (underutilization). Since ABR cell losses occur during overutilization periods of bandwidth mismatch cycles and the magnitude of bandwidth mismatch is proportional to the propagation delay, the effectiveness of a rate based flow control scheme decreases with increased propagation delay. The impact of increased propagation delay on the flow control schemes has been studied in the past [9], [10], [11], [12]. Besides the physical distance other system parameters namely the network, ABR and VBR time scales impact considerably on the effectiveness of the flow control schemes. In this paper, the impact of these system parameters on the effectiveness of the feedback based flow control is studied.

The network time scale is defined as the transmission time of a cell and it decreases as the network speed increases. Therefore, decreased network time scale corresponds to increased propagation delay. The VBR time scale is defined to be a measure of the rate of change of the VBR source rate and the ABR time scale is defined as the minimum distance among consecutive blocks of consecutive cells generated by the ABR source.

*This research is supported in part by the Advanced Research Project Agency under Grant F49620-93-1-0564 monitored by the Air Force Office of Scientific Research (AFOSR) and the National Science Foundation under Grant NCR-9628116.

The remainder of the paper is organized as follows. In section 2, the system considered in order to evaluate the impact of time scales is described. In section 3, a suitable Markov Chain for the system is formulated. Some numerical results are presented in section 4 and the paper ends with some conclusions presented in section 5.

2 Description of the System

In order to simplify the analysis and facilitate the understanding of the impact of the various time scales, a system with one VBR source and one ABR source is considered where each of these sources can deliver at most one cell per slot. The propagation delay between the ABR source and the network access node is assumed to be non-negligible. The ABR source rates are calculated at the network access node by using the current VBR source rate. The calculated transmission rates are fed back to the ABR source when a VBR source rate change occurs. Thus, the network access node is supposed to be able to detect a change in the VBR source rate. This may be possible through an explicit indication the VBR source carries by certain cells or an estimation mechanism implemented at the network access node.

The network access node is assumed to have a finite buffer of capacity C for the temporary storage of the ABR and VBR cells and the cell departures are assumed to occur before cell arrivals occurring over the same slot.

2.1 The ABR Traffic Source

As indicated previously, the ABR source rate is calculated based on the current VBR source rate. If the current VBR source rate is equal to r_v, then the maximum allowable ABR source rate will be equal to $1-r_v-\epsilon$. Here, ϵ is an arbitrarily small positive number to ensure that the maximum system load is below 1 (stability condition). It is assumed that the ABR source can always transmit with its maximum allowable rate (heavy traffic assumption). With this assumption, the worst case performance for the ABR source will be derived.

In order to satisfy its maximum allowable rate, the ABR source can transmit one cell per T_k slots where $T_k = \lceil \frac{1}{1-r_v-\epsilon} \rceil$. In this paper, T_k will be referred to as the *fundamental ABR subframe*. Note that, the same rate can also be achieved, if the ABR source transmits a batch with B_k cells per subframe of length $B_k T_k$ (*B_k-order subframe*).

The ABR time scale is the length of a B_k-order subframe ($B_k T_k$). Although the ABR time scale will be different under different order subframe transmission schemes, the same rate will be maintained. Figure 1 shows examples of transmission employing fundamental or higher-order subframes. Note that, for a given subframe, the ABR cells are transmitted only for the first B_k slots. In the rest of the paper, a fixed batch size (B_k) will be chosen for all possible ABR source rates and subframes. Thus, a static ABR time scale approach will be considered.

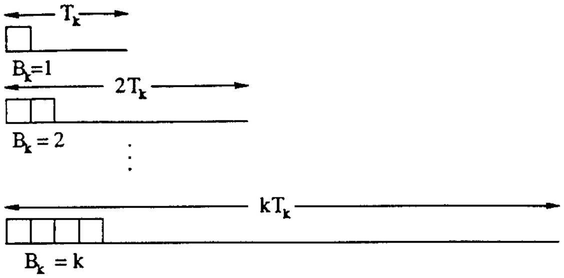


Figure 1: The ABR traffic subframes for a transmission rate of $1/T_k$

2.2 The VBR Traffic Source

Although the activity level of the VBR traffic source is not controllable and can change in principle at any slot, a change in the rate requires some time period to be manifested. In this paper, it will be assumed that such changes occur only at subframe boundaries for analysis tractability.

Let $\{S_k\}_{k\geq 1}$ denote a 2-state underlying Markov chain with state space $\overline{S} = \{0,1\}$. $\{S_k\}_{k\geq 1}$ will be used to describe the VBR source activity. Let $r_v(s_k)$ denote the rate of the VBR source at the k^{th} subframe, $s_k \in S$. Without loss of generality, it is assumed that $r_v(1) > r_v(0)$. The VBR cell arrival process is modeled as a Markov Modulated Bernoulli process. The probability of transmitting 1(0) cell in a slot of the k^{th} subframe is $r_v(s_k)(1-r_v(s_k))$.

Let $p_s(s_k)$ denote the probability that S_k changes at the end of the k^{th} subframe at the beginning of which it was in state s_k. $p_s(s_k)$ will reflect the average time over which VBR applications maintain a constant rate (or VBR time scale). If $p_s(0) = p_s(1) = p_s$, then the VBR time scale will be equal to $1/p_s$; otherwise, two time scales may be defined.

3 Analysis of the System

In order to determine the arrival process at the network access node over the k^{th} subframe, the network access node needs to know the current VBR source rate ($r_v(s_k)$) and whether the current ABR rate is based on the most recent feedback sent by the node or not (that is, whether the feedback's impact is not pending or pending, respectively). Therefore, let J_k be an indicator function which assumes the value of 1, if the impact of a feedback carrying a VBR source rate change is pending. If the impact of the feedback is not pending, J_k is equal to 0. The fundamental ABR subframe (thus, the ABR source rate) will be completely determined by (S_k, J_k) and it is given by $T_k = \lceil \frac{1}{1-r_v(s_k \oplus j_k)-\epsilon} \rceil$ where $\oplus$ denotes the modulo 2 addition.

A Markov chain can be constructed to describe the evolution of the system under fixed and non-zero roundtrip propagation delays. Here, a simplifying assumption is made by assuming that the roundtrip propagation delay is random and geometrically distributed with mean $\frac{1}{p_f}$, measured in terms of subframes.

Although it is possible that more than one feedback be pending at any time, at most one pending feedback will be considered in order to simplify the analysis. This approximation holds true if $max_{s_k}\{p_s(s_k)\} \ll p_f$; that is, if the VBR source time scale is much larger than the propagation delay. This is expected to be the case in order for a feedback based rate control scheme to be reasonably effective.

With these assumptions, it is easy to establish that the stochastic process $\{S_k, J_k, Q_k\}$ embedded at subframe boundaries is a Markov chain with state space $\{(s_k, j_k, q_k) : 0 \leq s_k \leq 1, 0 \leq j_k \leq 1, 0 \leq q_k \leq C\}$. Here, Q_k is a random variable describing the buffer occupancy at the beginning of the k^{th} subframe.

Let $P(s_k j_k q_k; s_{k+1} j_{k+1} q_{k+1})$ be the transition probability that the Markov chain $\{S_k, J_k, Q_k\}$ moves from state (s_k, j_k, q_k) to state $(s_{k+1}, j_{k+1}, q_{k+1})$. It is clear that (s_{k+1}, j_{k+1}) is independent from the queue occupancy (q_k) and thus $P(s_k j_k q_k; s_{k+1} j_{k+1} q_{k+1})$ can be expressed as follows.

$$P(s_k j_k q_k; s_{k+1} j_{k+1} q_{k+1}) = P(s_k j_k; s_{k+1} j_{k+1}) P(q_{k+1}/s_k j_k q_k)$$

where $P(s_k j_k; s_{k+1} j_{k+1})$ is the probability of passing from (s_k, j_k) to (s_{k+1}, j_{k+1}) and $P(q_{k+1}/s_k j_k q_k)$ is the probability of being at queue occupancy q_{k+1} at the beginning of the next subframe given the current subframe parameters. The derivation of these probabilities are shown in detail in [12].

After the calculation of the transition probabilities, the steady state probability distribution $\Pi(s_k, j_k, q_k)$ can be derived by using the following matrix equations, $\Pi = \Pi P$ and $\sum_i \Pi_i = 1$ where P is the state transition matrix each of whose elements show the transition probabilities $P(s_k j_k q_k; s_{k+1} j_{k+1} q_{k+1})$.

In order to study the impact of the network and source time scales, the ABR cell loss probability will be derived since the only negotiated QoS parameter for the ABR service class is the Cell Loss Ratio (CLR). The ABR cell loss probability can be calculated using the average number of cells lost and arrived in a subframe at the beginning of which $\{S_k, J_k, Q_k\}_{k\geq 1}$ is in state (s_k, j_k, q_k) and taking the expectation over all possible states.

Let, LP denote the ABR cell loss probability, and let $\overline{L}(s_k, j_k, q_k)$ and $\overline{M}(s_k, j_k, q_k)$ denote the number of cells lost and arrived in a subframe at the beginning of which $\{S_k, J_k, Q_k\}_{k\geq 1}$ in state (s_k, j_k, q_k) respectively. Then,

$$LP = E\{\frac{\overline{L}(s_k, j_k, q_k)}{\overline{M}(s_k, j_k, q_k)}\}$$

Let $I(s_k, j_k, q_k; i)$ $(R(s_k, j_k, q_k; i))$ denote the number of cells lost (arrived) at the i^{th} slot of the k^{th} subframe in state (s_k, j_k, q_k): $1 \leq i \leq B_k T_k$. Clearly, $R(s_k, j_k, q_k; i)$ does not depend on the queue occupancy and is only a function of s_k, j_k and i. Thus,

$$R(s_k, j_k, q_k; i) = R(s_k, j_k; i) = \begin{cases} A(s_k) & \text{if } i > B_k \\ A(s_k) + 1 & \text{if } i \leq B_k \end{cases}$$

where $A(s_k)$ denotes the number of VBR cell arrivals in a slot when the current VBR state is s_k.

The number of lost cells in a slot can be found by considering the number of arrivals in the slot, the buffer occupancy in the previous slot and the buffer capacity C. Therefore,

$$I(s_k, j_k, q_k; i) = \begin{cases} m & \text{if } R(s_k, j_k; i) = C + 1 - q_{k,i-1} + m \\ 0 & \text{otherwise} \end{cases}$$

where $q_{k,i-1}$ denotes the queue occupancy at the $i - 1^{th}$ slot of the k^{th} subframe.

Clearly, the number of cell arrivals over a subframe at the beginning of which the system Markov Process $\{S_k, J_k, Q_k\}_{k\geq 1}$ is in state (s_k, j_k, q_k) is the summation of $R(s_k, j_k; i)$ over all slots i. Thus, the average number of cell arrivals in a subframe at the beginning of which $\{S_k, J_k, Q_k\}_{k\geq 1}$ is in state (s_k, j_k, q_k) can be expressed as,

$$\begin{aligned} \overline{M}(s_k, j_k, q_k) &= \sum_{i=1}^{B_k T_k} E\{R(s_k, j_k; i)\} \\ &= B_k + B_k T_k r_v(s_k). \end{aligned}$$

The total number of cells lost over the entire subframe is equal to the summation of $I(s_k, j_k, q_k; i)$ for all i slots of the subframe. Thus, the following equation is obtained for $\overline{L}(s_k, j_k, q_k)$.

$$\overline{L}(s_k, j_k, q_k) = \sum_{i=1}^{B_k T_k} E\{I(s_k, j_k, q_k; i)\}$$

After some manipulations, $\overline{L}(s_k, j_k, q_k)$ can be shown to be equal to,

$$\begin{aligned} \overline{L}(s_k, j_k, q_k) &= \sum_{i=1}^{B_k} \sum_{l=0}^{C} \sum_{m=0}^{l-C+1} mP\{A(s_k) = C - l + m/q_k, s_k, j_k\} \\ &\times \quad P\{q_{k,i-1} = l/q_{k,0} = q_k\} \end{aligned}$$

Combining $\overline{L}(s_k, j_k, q_k)$ and $\overline{M}(s_k, j_k, q_k)$ the ABR cell loss probability can be expressed as follows.

$$\begin{aligned} LP &= \sum_{s_k=0}^{1} \sum_{j_k=0}^{1} \sum_{q_k=0}^{C} \frac{\Pi(s_k, j_k, q_k)}{B_k + r_v(s_k) B_k T_k} \\ &\times \sum_{i=1}^{B_k} \sum_{l=0}^{C} \sum_{m=0}^{l-C+1} mP\{A(s_k) = C - l + m/q_k, s_k, j_k\} \\ &\times \quad P\{q_{k,i-1} = l/q_{k,0} = q_k\} \end{aligned}$$

4 Numerical Results

In this section, some numerical results are presented to illustrate the impact of the network and source time scales on the performance of an ABR source which is located away from the network access node.

Figure 2 illustrates the ABR cell loss probability as a function of p_f for $p_s(0) = p_s(1) = p_s = 0.005, 0.001, 0.0005$ and

0.0001. For this plot, results are derived for VBR source rates $r_v(1) = 0.8$ and $r_v(0) = 0.4$ cells/slot and a buffer with capacity $C = 40$. As the propagation delay decreases (p_f increases), the amount of time that it takes for the ABR source to respond to the feedbacks generated at the network access node decreases. As a consequence, the amount of bandwidth mismatch (thus the overutilization periods) decreases, causing a decrease in the ABR cell loss probability. In addition to the propagation delay, the VBR time scale ($1/p_s$) also has a considerable effect on the ABR cell loss probability. When the VBR time scale decreases (p_s increases), VBR source rate changes occur more frequently. By this way, more bandwidth mismatch cycles are initiated resulting in more ABR cell losses. Note that $p_s \ll p_f$ in order for the assumption of no multiple pending feedbacks to be reasonably accurate.

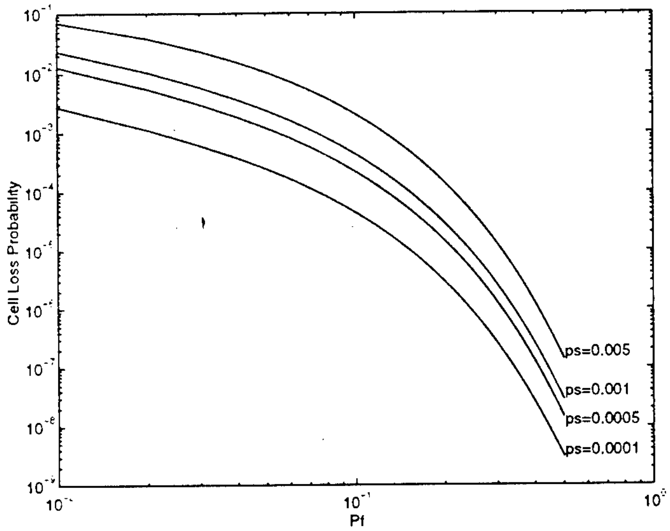


Figure 2: ABR cell loss probability vs. p_f for various p_s

Figures 3 and 4. illustrate the ABR cell loss probability as a function of batch size B_k for $C = 10, 15, 20, 25$ and 30. The distance between the ABR source and the network access node is assumed to be 20000 and 200 slots for Figure 3 and Figure 4 respectively. Since the VBR source rates are assumed to be $r_v(1) = 0.8$ and $r_v(0) = 0.4$, the fundamental subframe lengths are 2 and 6 slots. Thus, a B_k-order subframe (ABR time scale) has lengths of $2B_k$ and $6B_k$ slots. For a fixed value of C, it can be observed that, the ABR cell loss probability initially decreases as B_k (the ABR time scale) increases. This behavior is reversed when B_k exceeds a threshold. Thus, for a given C, there is an optimal B_k (or ABR time scale) that minimizes the induced ABR cell losses. The optimal B_k also depends on the propagation delay. For example, the optimal B_k for $C = 15$ is equal to 8 in Figure 3 while it is equal to 2 in Figure 4. This trend may be attributed to the decreasing positive impact of a large ABR time scale for a decreased propagation delay.

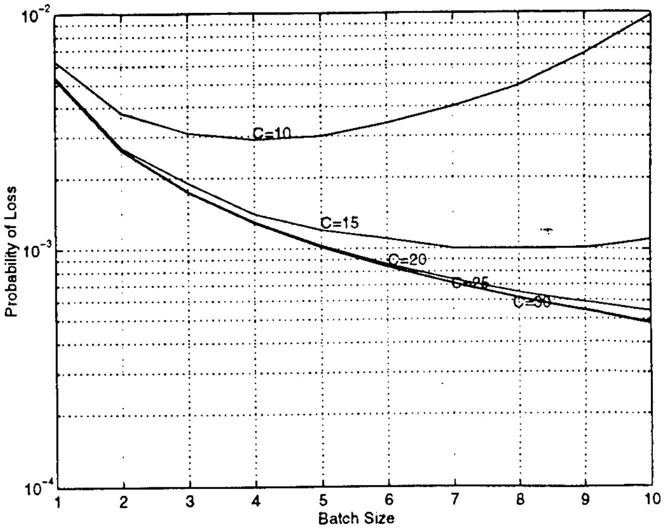


Figure 3: ABR cell loss probability vs. B_k for various values of C for an average distance of 20000 slots

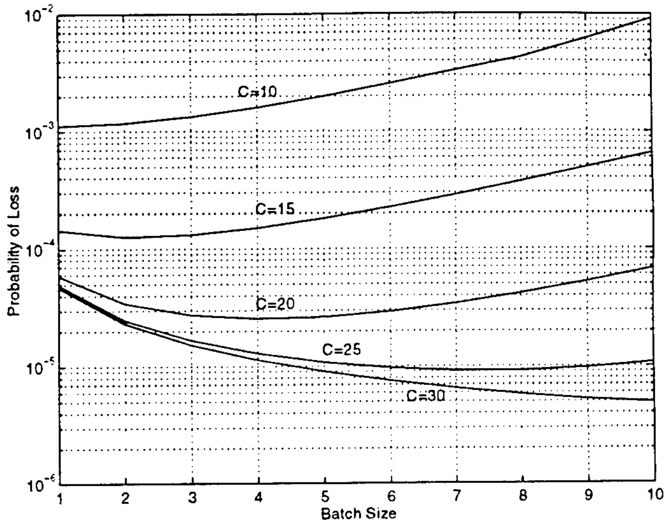


Figure 4: ABR cell loss probability vs. B_k for various values of C for an average distance of 200 slots

In addition to the network and source time scales, the VBR source rates also have considerable effects on the performance. Figure 5 shows the effects of the VBR source rates on the ABR cell loss probability. For a fair comparison, the mean VBR arrival rate is kept constant for the cases considered. The VBR source rates for the first case are $r_v(1) = 0.9$ and $r_v(0) = 0.3$ cells/slot, while for the second case $r_v(1) = 0.8$ and $r_v(0) = 0.4$ cells/slot. It can be clearly observed that the loss performance improves as the difference between the two rates decreases. For the case with $r_v(0) = 0.3$ and $r_v(1) = 0.9$, the fundamental subframe lengths are 2 and 11 slots. These subframe lengths correspond to ABR source rates of 0.5 and 0.091 respectively. Therefore, during a bandwidth mismatch initiated by a VBR source rate change, the total rate on the link can increase up to 1.4 causing potentially high losses. However, when VBR rates are $r_v(0) = 0.4$ and $r_v(1) = 0.8$, the corresponding ABR rates will be 0.5 and 0.167. During a bandwidth mismatch, the total rate on the link will be at most 1.3. Compared to 1.4, the overutilization in this case is decreased. Therefore, a decrease in the ABR loss probabilities occurs.

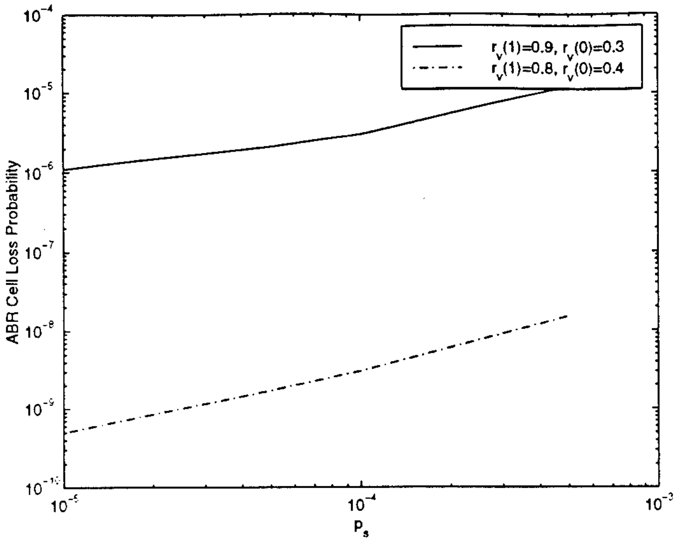


Figure 5: ABR cell loss probability vs. p_s for various VBR arrival rates

5 Conclusions

In this paper, the impact of network and source time scales on the performance of ABR applications is studied. The network time scale is defined as the transmission time of a cell. The VBR time scale is defined in terms of the rate of change of the VBR source rate. The ABR time scale is defined as the minimum distance between consecutive blocks of consecutive cells generated by the ABR source. In order to evaluate the impact of these time scales, a system with one ABR and one VBR source is considered.

While the impact of the network time scale has been considered in detail in the past, that of other relevant time scales has not been investigated. The main contribution of this paper is the study of the impact of the VBR and ABR time scales on feedback based flow control.

The increased VBR time scale has a positive impact on the ABR loss performance. This is due to the fact that, less frequent VBR rate changes (or bandwidth mismatch cycles) occur when the VBR time scale increases, resulting in reduced ABR cell losses. Increasing the ABR time scale (or the batch size B_k), initially decreases the loss probability. But, after the batch size exceeds a threshold, the situation is reversed. The optimal batch size -defined to be the one corresponding to this threshold- increases with the buffer capacity, as the capability of absorbing larger batches by the buffer increases with increased buffer space. For large propagation distances, the optimal batch size is large. As the propagation distance decreases, the optimal batch size decreases as well. This may be due to the reduced positive impact of the ABR time scale for decreased propagation delay.

The major conclusion of this work is that in addition to the network time scale, the ABR and VBR time scales may impact substantially a feedback based flow control as well. Thus the impact of *all system time scales* should be considered in order to accurately evaluate the effectiveness of a flow control algorithm.

References

[1] F. Bonomi and R. Morris, "The Rate Based Flow Control Framework for the Available Bit Rate ATM Service", *IEEE Network*, pp. 25-39, Mar./Apr. 1995.

[2] J. C. Bolet and A. Shankar, "Dynamical Behavior of Rate Based Flow Control Mechanism", *ACM Comp. Commun Rev.*, pp. 35-49, Apr. 1990.

[3] S. Liu et al., "Fairness in Closed Loop Rate Based Traffic Control Schemes", *ATM Forum/94-0387*, May 1994.

[4] N. Yin and M. G. Hluchjy, "On Closed-Loop Rate Control for ATM Cell Relay Networks," *INFOCOM'94*, pp.99-108, Toronto, 1994.

[5] R. Beraldi and S. Morano, "Selective BECN Schemes for Congestion Control of ABR traffic in ATM LAN," *ICC'96*, pp. 503-507, 1996.

[6] A. Barnhart, "Baseline Performance Using PRCA Rate Control," *ATM Forum/94-0597*, July 1994.

[7] A. Charny et al., "Congestion Control with Explicit Rate Indications," *ICC'95*, pp. 1954-1963, Seattle, 1995.

[8] L. Roberts, "Enhanced PRCA," *ATM Forum/94-0735R1*, Aug. 1994.

[9] Y. T. Wang and B. Sengupta, "Performance Analysis of a Feedback Congestion Control Policy Under non-negligible Propagation Delay," *ACM SIGCOMM'91*, pp. 149-157, Sep. 1991.

[10] M. Abdelaziz and I. Stavrakakis, "Study of an Adaptive Rate Control Scheme Under Unequal Propagation Delays," *ICC'95*, pp.1964-1968, Seattle, 1995.

[11] R. Pazhyannur and R. Agrawal, "Feedback Based Flow Control in ATM Networks with Multiple Propagation Delays," *1996 IEEE*, pp. 585-93.

[12] T. Dag and I.Stavrakakis, "Study of the Impact of Network and Source Time Scales on Feedback Based Flow Control," *SPIE'96*, Boston, Nov. 1996.

Earliest Due First Scheduling for Application-Level QoS Delivery*

Zoe Antoniou

Dept. of Electrical & Computer Engineering
Northeastern University
Boston, MA 02115

Ioannis Stavrakakis

Dept. of Electrical & Computer Engineering
Northeastern University
Boston, MA 02115

Abstract

The Quality of Service (QoS) of real time applications is affected in a rather direct manner by the timely delivery *of* application-level data units *which are typically mapped into several ATM cells. As a consequence, it would be reasonable to associate a* common *deadline with all the cells in one data unit (batch) and develop a scheduling policy that aims to maximise the number of data unit that are delivered by their respective deadlines.*

A batch-based Earliest Due First scheduling policy is proposed and analysed in this paper under periodic batch arrival processes which are typical of continuous media streams. Depending on whether partial batch service is performed or not, two variations of this scheduler are considered. It is shown that the proposed policies can be easily implemented in terms of a First Batch In - First Batch Out (FBI-FBO) and a Partial FBI-FBO schedulers respectively. Numerical results are presented and the induced batch and cell loss rates are compared against those induced by the standard FIFO scheduler as well as a FIFO scheduler which drops expired cells.

1 Introduction

Asynchronous Transfer Mode (ATM) technology is expected to support applications with a wide range of service requirements and traffic characteristics. Real time applications, such as voice and video are particularly demanding since such applications require stringent guarantees on the Quality-of-Service (QoS) in terms of bounded delay, loss and jitter. The traffic characteristics of real time applications can be very diverse as they combine a variety of media data such as text, audio, graphics, images and full motion video. The transmission of continuous media traffic is an inherently difficult problem due to the time sensitive nature and the traffic variability of these applications. [1] [6] [8]

Consider, for instance, the transmission of video applications in which it is required that frames (images) are transmitted at a certain rate in order to guarantee no starvation or overflow at the receiving end. Each frame would be of variable size (in bits) according to its complexity and the coding scheme employed. The frame could be considered as an application data unit which is central to both the traffic generation process at the transmitter and the reconstruction process at the receiver. Such a role may be played by a smaller encoding unit than the frame such as a macroblock. [7]

Application data units, such as the frame, are organised into small fixed size and standardised packets, the cells, in order to be transported over an ATM network. Typically, an application data unit is mapped into several ATM cells. When network resources are reserved at peak demand, all cells of the application data unit will be delivered on time, leading to a timely and error-free reconstruction at the receiver. When resources are over-allocated to a number of applications -to increase network utilisation- all cells of an application data unit may not be delivered before this application data unit is required by the reconstruction process. In this case, application data units may be significantly distorted due to missing cells or be impossible to process at all. The required QoS for such applications may then be defined in terms of a metric associated with the correctly and timely delivered application data units. An application-level QoS metric such as the application data unit loss rate would describe more accurately the QoS delivered to the end user, than a network-level QoS metric such as the cell loss rate. [2]

In order to achieve QoS delivery to real time applications while efficiently utilising the network resources it is necessary that allocation of the resources (scheduling) be based on the applications' QoS requirement. For instance, the timely delivery of information associated with real time applications suggests that deadlines could be defined and deadline-driven scheduling policies be designed. If the nature of an application requires that a complete application data unit should be received before it can be processed by the receiver, then it would be reasonable to associate a **common** deadline with all the cells in the same data unit and develop scheduling policies that aim to maximise the number of data unit that are delivered by their respective due date. This approach is different from many traditional approaches in which every cell is assigned its own deadline and the scheduling policy

*Research supported in part by the Advanced Research Project Agency under Grant F49620-93-1-0564 monitored by the Air Force Office of Scientific Research (AFOSR) and the National Science Foundation under Grant NCR 9628116.

typically attempts to maximise the number of individual cells transmitted by their due date.

In this paper a scheduling policy which would be appropriate for supporting applications such as those described above is introduced. This policy attempts to maximise the number of application data units - referred to as batches in the rest of the paper- served before their deadlines.

Unlike the typical cell-based Earliest Due First (EDF) scheduling policy which requires sorting on a cell-by-cell basis [4], the proposed policy requires such sorting on a batch-by-batch basis, reducing the processing burden. For continuous media applications in which batches (application data units) are delivered over well defined periods the processing burden of the proposed policy can be substantially reduced by ordering the multiplexed data streams according to their respective deadlines and serving them according to the First Batch In - First Batch Out (FBI-FBO) policy, as it will become clear in later sections. The complexity of this policy is similar to that of a (cell-based) FIFO policy applied to the deadline ordered streams. The performance of the FBI-FBO and the FIFO policies is evaluated in terms of the induced cell loss and batch loss probabilities. In view of the earlier discussion the batch loss probability may be a more relevant metric for applications in which higher layer protocol data units are fragmented into a number of ATM cells; these data units may be considered to be lost if any one of their cells are lost, unless some redundancy is introduced in the data unit to improve the probability of correct reception at the expense of increasing the traffic in the network.

2 System Description

This section describes the system and the traffic environment considered in this work. A network node with N incoming and one outgoing links is considered. All links are assumed to have the same capacity and, thus, a cell requires a fixed amount of time for transmission, referred to as a slot. All arrivals and departures from the node occur at slot boundaries. Let J denote an incoming data stream (application); J takes values in the range $0 \leq j \leq N-1$. A variable with superscript j represents a quantity associated with stream j.

The incoming traffic is assumed to be bursty periodic. It consists of variable size bursts of cells arriving at fixed time intervals (periods); all applications are assumed to have the same period, T. An example of a batch arrival pattern can be seen in Figure 1. This type of traffic is characteristic of many continuous media applications such as digital audio and video. It is further assumed that the bursts of stream j arrive over consecutive time slots, that is in the form of batches, and that their length, A^j, follows a general distribution with expected value $\overline{A^j}$. All batches of the same application j are assumed to have the same deadline, D^j, which is provided to the server upon connection set up. The *deadline* determines the maximum time (measured in cell slots) which the batch may remain in the node once the header of the batch has arrived. This includes both the delay and the transmission delay.

Figure 2 depicts the instants at which batch arrivals occur for $N = 4$ streams. Let $\{t_k\}_{k\geq 0}$ denote the sequence of batch arrival instants (from any stream); t_k indicates the arrival instance of the k^{th} batch. Variables with subscript k will represent the value of the associated quantity at time t_k. Let τ_k denote the interval between two consecutive batch arrivals (from any stream); that is :

$$\tau_k = t_{k+1} - t_k \qquad \text{with} \qquad (1)$$

$\tau_k = \tau^j$ if $J_k = j$, where τ^j denotes the interarrival time between two consecutive batches from streams j and $(j+1)mod(N)$, or equivalently, the distance between two consecutive batch arrival instants of streams j and $(j+1)mod(N)$. In the rest of this paper it is assumed that the streams are ordered in terms of increasing deadlines. That is, $D^j \leq D^{j+1}$ for $0 \leq j \leq N-2$.

As it was stated earlier, the goal of the proposed policy is the timely delivery of complete batches. This goal suggests that the scheduler should serve batches based on the Earliest Due First (EDF) policy. In view of the deadline ordering and the periodicity of the batch arrival process it is easy to establish that a scheduler that serves *batches* according to the *First In First Out* policy does serve them according to the EDF policy. This proposed policy is referred to as the First Batch In - First Batch Out (FBI-FBO) policy and is discussed in more detail below.

2.1 The FBI-FBO Scheduling Policy

Implementation of this policy requires that information regarding the length (in cells) of a batch be available to the scheduler upon batch arrival. This will be the case if the batch size is fixed and provided to the scheduler upon connection set up or if this information is carried by the first cell of the batch or by dedicated cells such as Resource Management Cells.

Upon Batch arrival the scheduler decides whether this batch can be served by its deadline or not. In the former case, the batch is served before its deadline expires as described below. In the latter case, the *entire* batch is discarded.

The scheduler is equipped with a *service queue* to which cells are shifted for storage before they are transmitted. All cells shifted to the service queue will be transmitted, following the first cell in - first cell out policy. Suppose that a batch from stream j arrives at time t_k. If the sum of the batch length A^j and the service queue occupancy Q does not exceed the batch deadline, D^j, a reservation is made in the service queue for a space of A^j slots. The service queue occupancy is then registered as being $(Q + A^j)$, even though only the first cell has arrived so far. The remaining $(A^j - 1)$ cells of the batch will be arriving over the next $(A^j - 1)$ time slots and they will be shifted to the reserved space in the service queue immediately upon arrival. In other words, data stream j 'sees' a logical service queue of capacity D^j and decides to initiate a batch shifting to the service queue only if the

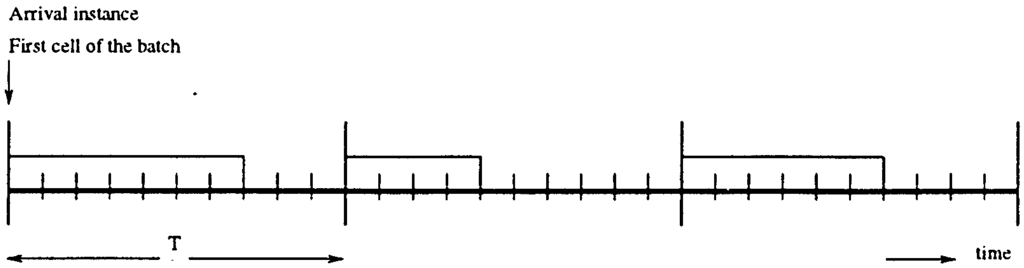


Figure 1: An example batch arrival pattern.

following condition is met upon arrival:

$$Q + A^j \leq D^j \tag{2}$$

This condition guarantees that the cells will be sent to the output link within D^j time slots and, thus, they will meet their deadline. If the associated logical queue overflows, then the dropped cells will be the ones to miss the batch deadline. In this case no cell of the batch is shifted to the service queue.

It should be noted that the server attends to a newly arrived batch once upon the arrival of the header, determines whether it will meet its deadline or not and decides whether to serve it or drop it. Once a scheduling decision is made, it is no longer necessary to keep a record of the batch arrival, its length or its relative deadline. The above scheduling policy is *work conserving* in the sense that the server will not remain idle as long as there are batches waiting to be served. If the service queue becomes empty, the server will become idle and will await for the next batch arrival.

2.2 The Partial FBI-FBO Scheduling Policy

In an environment in which complete batches need to be transmitted before their deadline in order to be useful, a scheduling policy that drops expired cells is more effective than the one which serves expired cells, in the sense that a larger fraction of batches can be transmitted before their deadlines. In order for the scheduler to determine whether an entire batch will meet its deadline or not, information about the length of the batch must be provided upon batch arrival. This is the assumption made in the FBI-FBO policy described above.

If the batch length information is not provided upon batch arrival the scheduler would attend to each cell of the batch individually by checking the service queue occupancy against the remaining time till the deadline expiration on a cell-by-cell basis. In this case, though, the scheduler may serve part of a batch and drop the remaining cells which will not meet the deadline. The entire batch would then be lost. The variation of the FBI-FBO policy -referred to as Partial FBI-FBO policy- is considered in this paper as well. This is similar to the FBI-FBO policy with the difference that the server serves the fraction of the batch that can meet its deadline. Clearly, a smaller number of complete batches and a higher number of cells would be served under the Partial FBI-FBO policy.

3 Performance Analysis

In this section a more precise notation is introduced and the derivation from first principles of the equations that mathematically describe the system is presented.

Let J_k denote the data stream from which a new batch arrival occurs at time t_k. The content of the new arrival is to be shifted to the service queue beginning at time t_k. Let Q_k denote the service queue occupancy at time t_k, *after* it has been updated by the number of cells in the new batch from stream J_k. In view of the above, the system process $\{Q_k, J_k\}_{k \geq 0}$ can be defined as a 2-D process with state space $\{(q_k, j_k) : 0 \leq q_k \leq D_{max}, 0 \leq j_k \leq N-1\}$. D_{max} denotes the maximum deadline of all N stream deadlines; that is $D_{max} = \max_j\{D^j\}$. Note that for J_k the only transitions allowed are from some j_k to $j_{k+1} = (j_k + 1) mod(N)$. Based on the above definition of the system process, it is easy to establish that it is a Markov chain. This can be established by noting that J_k is a periodic random variable and that the next service queue occupancy, Q_{k+1}, can be probabilistically determined from the current one, Q_k, and the new batch arrival, $A^{j_{k+1}}$; A^{j_k} denotes the random variable representing the number of cells from stream j_k that arrived at time t_k. The remaining of this section contains the derivation of the probability that the Markov chain moves from any state (q_k, j_k) to state (q_{k+1}, j_{k+1}), given by $P(q_k, j_k, q_{k+1}, j_{k+1})$.

3.1 FBI-FBO Policy

Under this policy, cells of a batch are served only if the *entire* batch can meet its deadline. Three distinct cases can be identified, based on the queue occupancy of the service buffer at the instant of a new batch arrival. In each of the following cases $0 \leq j_k \leq N-1$ and $j_{k+1} = (j_k + 1) mod(N)$.

Case A: for $0 \leq q_k \leq \tau_k$. In this case all the cells in the buffer will have been served by the time the next batch arrives from stream j_{k+1} at time t_{k+1} and the new buffer occupancy will be equal to the number

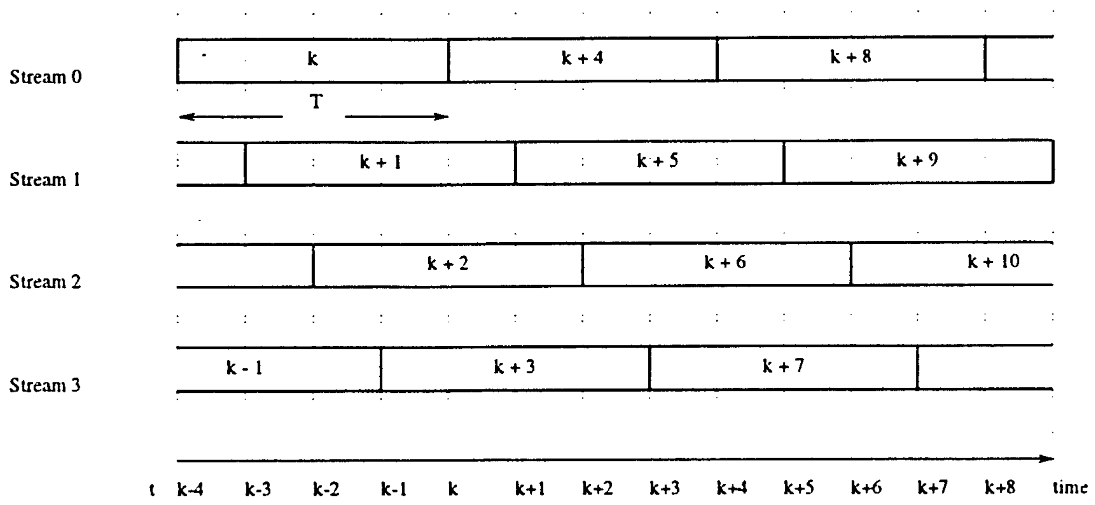


Figure 2: Periodic arrival pattern with shifted relative arrival instants for $N = 4$ traffic streams.

of new arrivals only:

$$P(q_k, j_k, q_{k+1}, j_{k+1}) = P\{A^{j_{k+1}} = q_{k+1}\}$$

$$\text{for} \quad 0 \leq q_{k+1} \leq T \qquad (3)$$

Case B: for $\tau_k < q_k \leq D^{j_{k+1}} + \tau_k - 1$. In this case by the time the next batch arrives the service buffer occupancy will have dropped to $(q_k - \tau_k)$. Hence, with the new arrival $A^{j_{k+1}}$ the occupancy will be updated to $(q_k - \tau_k + A^{j_{k+1}})$ if the entire batch can meet its deadline $D^{j_{k+1}}$:

$$P(q_k, j_k, q_{k+1}, j_{k+1}) = P\{A^{j_{k+1}} = q_{k+1} - (q_k - \tau_k)\}$$

$$\text{for} \quad (q_k - \tau_k) \leq q_{k+1} \leq D^{j_{k+1}} \qquad (4)$$

Otherwise, the new batch will be dropped and the service buffer occupancy will become $(q_k - \tau_k)$:

$$P(q_k, j_k, q_{k+1}, j_{k+1}) = \sum_{n=D^{j_{k+1}}-(q_k-\tau_k)+1}^{T} P\{A^{j_{k+1}} = n\}$$

$$\text{for} \quad q_{k+1} = q_k - \tau_k \qquad (5)$$

Case C: for $D^{j_{k+1}} + \tau_k \leq q_k \leq D_{max}$. In this last case the new batch will definitely miss its deadline and thus it is dropped with probability 1:

$$P(q_k, j_k, q_{k+1}, j_{k+1}) = 1 \quad \text{for} \quad q_{k+1} = q_k - \tau_k \quad (6)$$

3.2 Partial FBI-FBO Policy

This policy allows for a batch to be either partially or fully served. In the event where only a fraction of the batch can meet its deadline those cells are shifted to the service buffer and the remaining are dropped. Again three distinct cases can be identified. Cases A and C are the same as in the above section.

Case B: for $\tau_k \leq q_k \leq D_{j_{k+1}} + \tau_k - 1$. In this case depending on the length of the new batch there are two distinct possibilities of full or partial service:
Full service:

$$P(q_k, j_k, q_{k+1}, j_{k+1}) = P\{A^{j_{k+1}} = q_{k+1} - (q_k - \tau_k)\}$$

$$\text{for} \quad (q_k - \tau_k) \leq q_{k+1} \leq D^{j_{k+1}} - 1 \qquad (7)$$

Partial service:

$$P(q_k, j_k, q_{k+1}, j_{k+1}) = \sum_{n=D^{j_{k+1}}-(q_k-\tau_k)}^{T} P\{A^{j_{k+1}} = n\}$$

$$\text{for} \quad q_{k+1} = D^{j_{k+1}} \qquad (8)$$

4 Computation of Batch Loss and Cell Loss probabilities

Apart from the *cell* loss probability that can be used in order to assess the performance of the proposed policies, another meaningful metric would be the *batch* loss probability. A batch is considered to be lost unless all its cells are served. The batch or cell loss probabilities are easily computed by involving the Markov system process.

4.1 Batch Loss Probability

Since a batch is considered lost if one of its cells is lost, the analysis for the batch loss probability is the same for both policies.

Let $L^B(q_k, j_{k+1})$ denotes the number of batches that are dropped over the interval $[t_{k+1}, t_{k+2})$, where j_{k+1} is the data stream the scheduler will attend to over this interval (following stream j_k) and q_k is the queue occupancy at t_k. Clearly:

$$L^B(q_k, j_{k+1}) = 1_{\{A^{j_{k+1}} > D^{j_{k+1}} - (q_k - \tau_k)\}} \tag{9}$$

The conditional expected value of this indicator function given that the system is in state (q_k, j_{k+1}) is given by:

$$\overline{L^B}(q_k, j_{k+1}) = E\{1_{\{A^{j_{k+1}} > D^{j_{k+1}} - (q_k - \tau_k)\}}\} \tag{10}$$

It represents the average number of batches lost as a function of the queue occupancy at the time instant t_k and the data stream arrival that occurs at t_{k+1} under both policies.

If $0 \leq q_k \leq \tau_k$ then clearly there will be no losses since the server will serve all the cells in the service buffer during the interarrival time τ_k between streams j_k and j_{k+1} and thus:

$$L^B(q_k, j_{k+1}) = 0 \tag{11}$$

$$\overline{L^B}(q_k, j_{k+1}) = 0 \tag{12}$$

If $\tau_k \leq q_k \leq D_{max}$ then losses may occur. The conditional expected value of those losses in any state (q_k, j_{k+1}) is given by:

$$\overline{L^B}(q_k, j_{k+1}) = \sum_{n=D^{j_{k+1}}-(q_k-\tau_k)+1}^{T} P\{A^{j_{k+1}} = n\} \tag{13}$$

Hence, the overall average value of the batch losses L^B over any interval τ_k will be:

$$\overline{L^B} = \sum_{j=0}^{N-1} \sum_{q_k=\tau_k+1}^{D_{max}} \sum_{n=D^{j_{k+1}}-(q_k-\tau_k)+1}^{T} P\{A^{j_{k+1}} = n\}\pi(q_k, j_k), \tag{14}$$

where $\pi(q_k, j_k)$ is the steady state probability distribution of the corresponding Markov Chain.

Finally, the batch loss probability is given by:

$$P_B = \frac{E\{\text{batches lost over any interval } \tau_k\}}{E\{\text{batches arrived over any interval } \tau_k\}} = \frac{\overline{L^B}}{1} = \overline{L^B} \tag{15}$$

The denominator is equal to 1 since by definition of the system, there is only one batch arrival over any interarrival interval τ_k.

4.2 Cell Loss Probability

The derivation of the cell loss probabilities is similar for both policies and follows the same procedure as in the above section. Under the FBI-FBO policy, the cell losses correspond to the cells of the batches which did not meet the deadline. Whereas under the Partial FBI-FBO policy the lost cells can be a fraction of the batch's total number of cells.

Again let $L^C(q_k, j_{k+1})$ denote the number of cells dropped over the interval $[t_{k+1}, t_{k+2})$, where j_{k+1} is the data stream the scheduler will attend to over this interval and q_k is the queue occupancy at t_k. Also, let $\overline{L^C}(q_k, j_{k+1})$ denote the conditional expected value given that the system is in state (q_k, j_{k+1}).

As in the previous section, if $0 \leq q_k \leq \tau_k$ then clearly there will be no losses since the server will serve all the cells in the service buffer during the interarrival time τ_k between streams j_k and j_{k+1} and thus:

$$L^C(q_k, j_{k+1}) = 0 \tag{16}$$

$$\overline{L^C}(q_k, j_{k+1}) = 0 \tag{17}$$

for both policies.

If $\tau_k \leq q_k \leq D_{max}$ then losses may occur. For the FBI-FBO policy the conditional average number of cells lost over any interval τ_k is given by:

$$\overline{L_1^C}(q_k, j_{k+1}) = \sum_{n=D^{j_{k+1}}-(q_k-\tau_k)+1}^{T} nP\{A^{j_{k+1}} = n\} \tag{18}$$

The same quantity for the Partial FBI-FBO policy is given by:

$$\overline{L_2^C}(q_k, j_{k+1}) = \sum_{n=D^{j_{k+1}}-(q_k-\tau_k)+1}^{T} [n-(D_{j_{k+1}}-(q_k-\tau_k))]P\{A^{j_{k+1}} = n\} \tag{19}$$

The overall average values of $L_1^C(q_k, j_{k+1})$ and $L_2^C(q_k, j_{k+1})$ is given by:

$$\overline{L_1^C}(q_k, j_{k+1}) = \sum_{j=0}^{N-1} \sum_{q_k=\tau_k+1}^{D_{max}} \sum_{n=D^{j_{k+1}}-(q_k-\tau_k)+1}^{T} nP\{A^{j_{k+1}} = n\}\pi(q_k, j_k) \tag{20}$$

$$\overline{L_2^C}(q_k, j_{k+1}) = \sum_{j=0}^{N-1} \sum_{q_k=\tau_k+1}^{D_{max}} \sum_{n=D^{j_{k+1}}-(q_k-\tau_k)+1}^{T} P\{A^{j_{k+1}} = n\}[n-(D^{j_{k+1}}-(q_k-\tau \tag{21}$$

where $\pi(q_k, j_k)$ is the steady state probability distribution of the corresponding Markov Chain.

Finally, the cell loss probability is given by:

$$P_C = \frac{E\{\text{cells lost over any interval } \tau_k\}}{E\{\text{cells arrived over any interval } \tau_k\}} \tag{22}$$

The numerator is given by Equations (20) and (21). The denominator represents the average number of cells that arrive in the system over any interval τ_k. It can be computed by summing the mean batch length of each data stream weighted by the probability that a batch that arrives in any given interval τ_k belongs to a particular data streams. Since all the data streams have the same frequency of batch arrivals in a period the weighted sum over any interval τ_k is given by:

$$\overline{A} = \sum_{j=0}^{N-1} \frac{1}{N} \overline{A^j} \qquad (23)$$

5 Numerical Results and Discussion

In this section, the performance of the proposed policies is compared to that of the FIFO policy which serves the deadline ordered streams on a cell-by-cell basis. Comparison is based on numerical results for the batch loss and cell probabilities induced by these policies.

The induced losses under the proposed policies have been computed using equations (14), (15), (20), (21), (22) and (23) while those under the FIFO policy have been computed through simulations. For each simulation run 10^7 batch instants have been generated from each stream.

Two versions of the FIFO policy have been considered: FIFO policies with and without an expired cell discarding mechanism. The first one assumes no knowledge of the deadlines and simply serves the cells on a first-come first-served basis. This is referred to as the pure FIFO approach in this paper and it is simple and straightforward in its implementation. The proposed policy is compared to the pure FIFO in order to evaluate their relative performances and to determine if a significant gain can be achieved by employing a more involved scheduling policy. The second FIFO policy that is considered has an expired cell discarding mechanism. It identifies and drops those cells that can not meet their deadline. In order for the expired cells to be dropped they typically need to be searched for. The computational overhead in this case is quite significant for a large number of streams. Another drawback of the FIFO approach is that cells from different batches are multiplexed at the service queue. This causes the shape of the data traffic to change and one information unit spreads over a longer time interval in the output link as compared to its original length.

5.1 Homogeneous Traffic

Initially a homogeneous traffic environment is considered whereby the traffic from all data streams is statistically identical and has the same performance requirement (that is common delay tolerance). The batch lengths are geometrically distributed. The interarrival interval τ_k is constant and equal to $\tau = \frac{T}{N}$ (τ is always rounded to the nearest integer). The system utilisation can be varied in proportion to the average batch length and the period T.

Figure 3 shows the batch loss probability under the FBI-FBO policy (referred to here as Policy 1), the Partial FBI-FBO policy (referred to here as Policy 2) and the FIFO with the expired cell discarding mechanism as a function of the system utilisation for $N = 4$, $T = 40$ and identical deadlines $D^j = D$. The following deadlines have been considered: $D = 40$ (solid line) , $D = 60$ (dotted line) and $D = 80$ (dashed line). In other words, the deadline has been set to 1, $1\frac{1}{2}$ and 2 frames (periods). Note that as the deadlines increase the losses decrease as was expected.

Figure 4 shows how the batch loss probability changes as a function of the offered load for all four policies for $N = 4$ and $T = 40$. The results are for deadlines equal to 1 and $1\frac{1}{2}$ frames. The top line with x-marks corresponds to the FIFO without the expired cell discarding mechanism, the line with stars corresponds to the FIFO with expired cell discarding, the line with circles corresponds to Policy 2 and the bottom solid line is Policy 1. Among the four policies, Policy 1 provides the lowest batch loss rate. For low system utilisation it performs better than the FIFO approaches by approximately an order of magnitude. As the system utilisation increases Policy 2 and the FIFO with the expired cell discarding mechanism tend to converge to a common batch loss rate whereas the losses of the pure FIFO increase at a faster rate.

Similar observations can be made when the number of data streams is increased to $N = 8$. Figure 5 shows the results for deadlines equal to 1 and $1\frac{1}{2}$ frames and $T = 64$. Again the policies at the two extreme ends are Policy 1 and the pure FIFO. Policy 1 provides the lowest batch loss rate among the four and outperforms the pure FIFO by at least one order of magnitude. Policy 2 and the FIFO with the expired cell discarding mechanism tend to converge for high system utilisations.

So far, only the batch loss rates have been considered. Figure 6 shows the cell loss rates for the cases considered in Figure 5. Notice that Policy 2 (line with circles) performs better than Policy 1 (lower solid line) as expected. However, the performance of Policy 1 is the same or better than that of the pure FIFO (upper line with x-marks).

From the above results it can be clearly concluded that Policy 1 has the lowest batch loss rate while at the same time it keeps the cell loss rate at the same or lower level than that of the pure FIFO approach. The performances of Policy 2 and of the FIFO with the expired cell discarding mechanism are generally in the middle with higher batch loss rates and slightly lower or equal cell loss rates than Policy 1.

Increasing the deadline of the input traffic to 2 frames or more (not shown in this paper) resulted in about the same performance under all policies. In other words, if the deadlines are long enough the losses become independent of the particular scheduling policy and they depend on the traffic load alone.

Figures 7 and 8 present the results under all four policies under a batch size distribution delivering a batch size of length L cells with probability p and length 1 with probability $1 - p$ ($1 - L$ bursty periodic traffic model). The system parameters are: $N = 8$, $T = 64$, $L = 36$, $p = 0.2$ and deadlines equal to 1 and

$1\frac{1}{2}$ frames.

Figure 7 shows the batch loss probability as a function of the offered load. Once again Policy 1 achieves the lowest batch loss rate whereas the pure FIFO policy has the highest.

The cell loss probability is shown in Figure 8. As it can be seen, Policy 1 induces equal or lower cell loss rate than the pure FIFO approach, as before. Again the losses spread over mostly three orders of magnitude as opposed to five or six orders under geometrically distributed batches.

5.2 Inhomogeneous Traffic

An inhomogeneous traffic environment has also been considered. The objective in this case is to evaluate the level of diversity in the induced performance. Out of the $N = 8$ data streams, two are assumed to deliver probabilistically larger batches and since the period is the same for all streams, have larger load. τ is maintained constant for all streams. The six (background) streams are assumed to deliver batches of size 1 or L with parameter p (as before) with $L = 10$ and $p = 0.2$. The two (more active streams) streams have parameters L and $p = 0.4$, where L assumes a value between 8 and 56 so that a specific system utilisation is achieved.

Figure 9 shows the batch loss rates for the two more active streams (data streams 1 and 4) and the six background streams under all four policies. It can be observed that Policy 1 induces the lowest losses for the two more active streams which are of the order of $10^{-2} - 10^{-3}$ and are observed only for a system load higher than 0.7. The other policies induce higher losses for a wider range of system utilisation. The two FIFO policies induce approximately the same losses which are higher than those under Policy 1 by at least an order of magnitude. For the six background streams Policy 1 induces a loss rate of the order of $10^{-3} - 10^{-4}$ for system utilisation higher than 0.7 only.

To further reduce the batch losses an attempt was made to adjust the streams' interarrival time τ in proportion to the *expected batch length* of each stream. Increasing the values of τ for streams with high mean batch length could result in a reduction of the average occupancy of the service buffer since 'busy' streams will now have more time to be served before the next batch arrival occurs. The value of τ has been set using the following equation:

$$\tau^j = \frac{\overline{A_T^j}}{\sum_{j=0}^{N-1} \overline{A_T^j}} T \tag{24}$$

and is rounded to the nearest integer as before.

Figure 10 shows the batch losses for Policy 1 with equal values of τ (solid line) and with weighted values of τ (dashed line). It can be noted that adjusting the value of τ hardly affects the loss rate for the two more active streams. In fact, there is a slight increase. On the other hand, the loss rate for the six background has been improved significantly. Losses are of the order of 10^{-5} only for system utilisations higher than 0.9.

The more active streams have longer batches as compared to other streams. Long batches will reserve more space in the service buffer once they arrive and are accepted by the system. Suppose that a more active stream is followed by a few less active ones which is followed in turn by a more active one and so on. This is the case for the batch losses in Figure 10. Once a long batch is accepted by the system it will be shifted to the service buffer. The longer the interarrival interval τ is the more cells will be served before the next batch arrives thus creating more free slots in the service buffer for the next arrivals. This implies that fewer batches will be dropped from the less active streams which follow (decrease in the loss rate for the less active streams as seen in Figure 10). On the other hand, if the interarrival interval τ is the same for all streams then shorter batches from less busy streams may have to be dropped more frequently.

References

[1] R.P. Tsang et al., "Dynamic Resource Control for Continuous Media Traffic over ATM Networks". *Computer Communications*, v. 19, 1996, pp. 1092-1111.

[2] S. S. Lam and G.G Xie, "Group Priority Scheduling". Technical Report TR-95-28, University of Texas at Austin, Austin, Texas, January 1996.

[3] S.S. Lam and G.G. Xie, "Real-Time Block Transfer Under a Link Sharing Hierarchy". Technical Report, University of Texas at Austin, Austin, Texas, 1996.

[4] X. Wang and I. Stavrakakis, "Study of Scheduling for Group-based Quality of Service Delivery". *Third volume on Performance Modeling and Evaluation of ATM Networks*, ed.: D. Kouvatsos, Chapman & Hall, 1997.

[5] I.R. Philp, K. Nahrstedt and J.W.S. Liu, "Scheduling and Buffer Management for Soft-Real-Tile VBR Traffic in Packet-Switched Network". Technical Report, University of Illinois at Urbana-Champaign, Ill.

[6] K. Nahrstedt, "Challenges of Provided End-to-End QoS Guarantees in Networked Multimedia Systems". Technical Report, University of Illinois at Urbana-Champaign, Ill.

[7] J.M. Mc Manus and K.W. Ross, "Video on Demand over ATM: Constant-Rate Transmission and Transport". Technical Report, University of Pennsylvania, Philadelphia, PA, November 1995. http://www.seas.upenn.edu/~ross/.

[8] V. Srinivasan, A. Ghanwani and E. Gelenbe, "Block Loss reduction in ATM Networks". *Computer Communications*, v. 19, 1996, pp. 1077-1091.

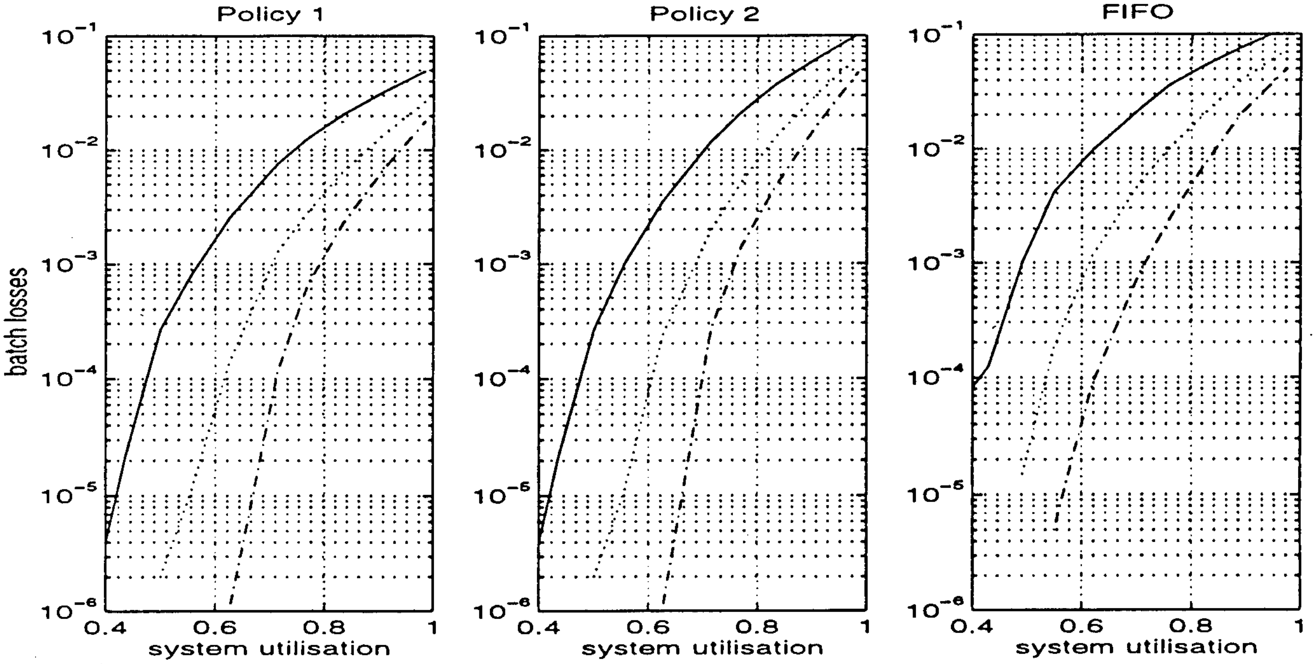


Figure 3: Batch loss curves for N=4 streams, period T=40 and deadlines D=40(solid), 60(dotted), 80(dashed line) for HOMOGENEOUS periodic traffic with GEOMETRIC distribution. The losses decrease as a function of increasing deadlines as expected.

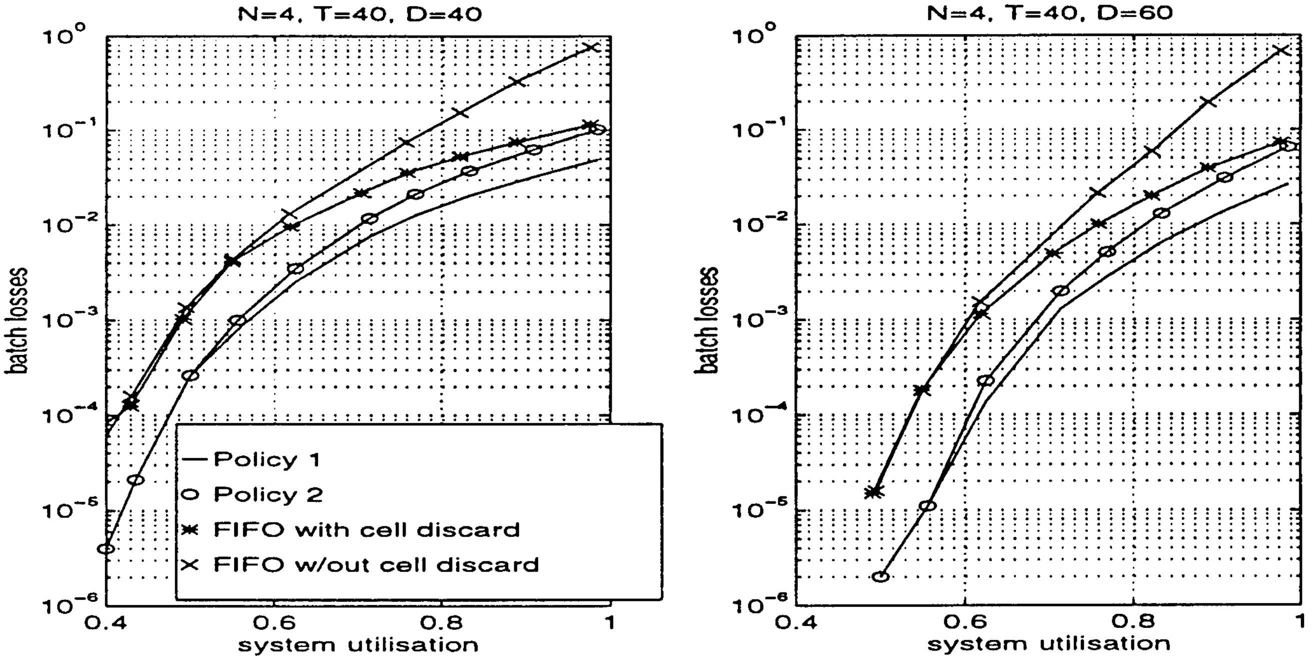


Figure 4: Batch loss curves for N=4 streams, period T=40 and deadlines D=40 and 60 (GEOMETRIC distribution). Top line is FIFO without expired cell discarding, line with the stars is FIFO with expired cell discarding, line with circles is Policy 2 and bottom solid line is Policy 1.

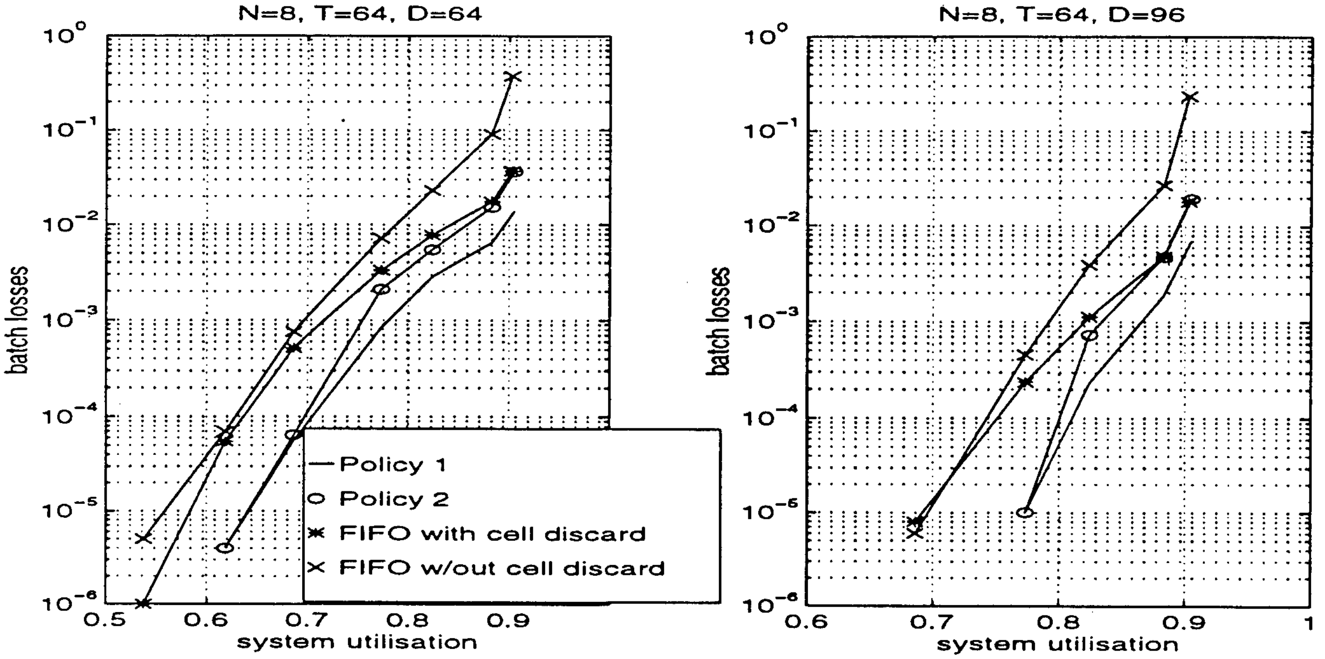


Figure 5: Batch loss curves for N=8 streams, period T=64 and deadlines D=64 and 96 (GEOMETRIC distribution). Top solid line is FIFO without expired cell discarding, dashed line is FIFO with expired cell discarding, dotted line is Policy 2 and bottom solid line is Policy 1.

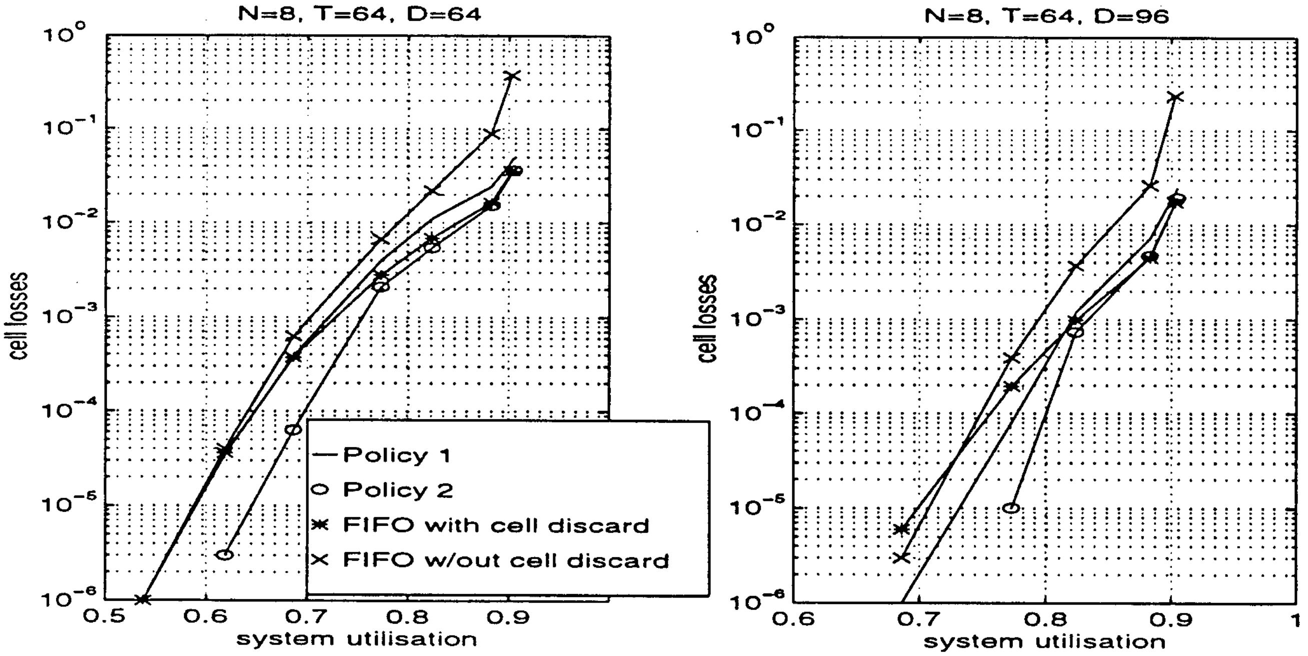


Figure 6: Cell loss curves for N=8 streams, period T=64 and deadlines D=64 and 96 (GEOMETRIC distribution). Top line is FIFO without expired cell discarding, line with stars is FIFO with expired cell discarding, line with circles is Policy 2 and bottom solid line is Policy 1.

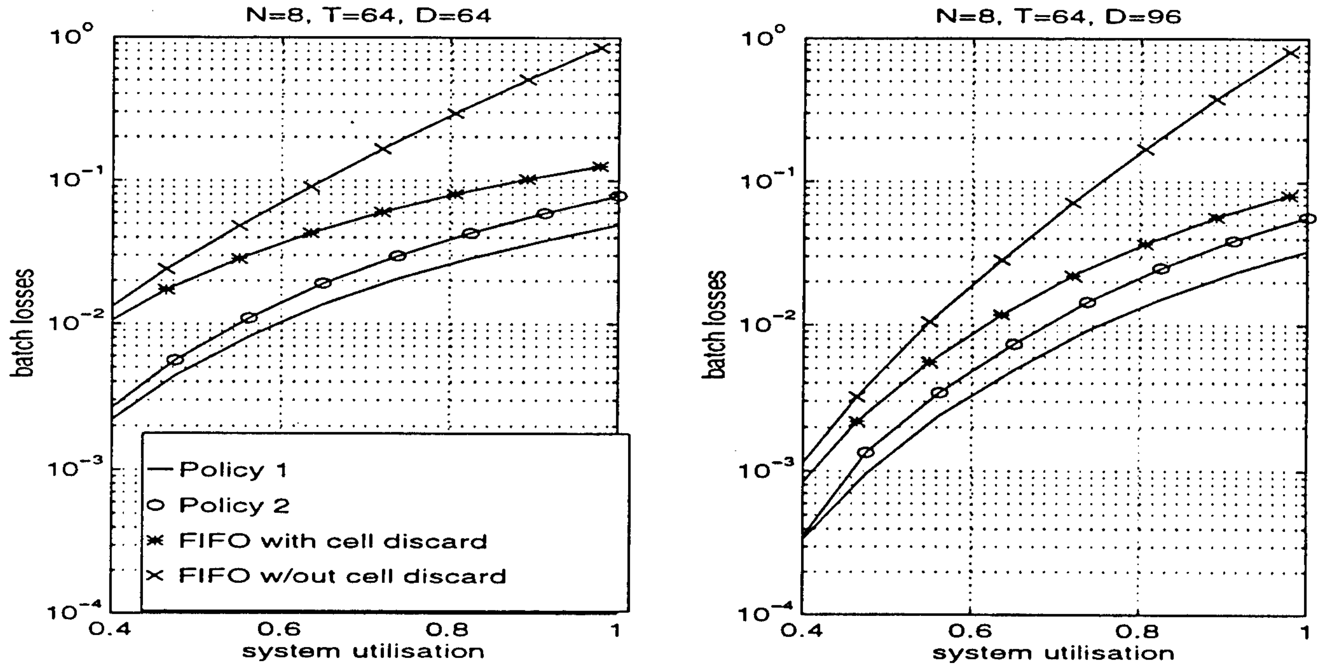


Figure 7: Batch loss curves for N=8 streams, period T=64 and deadlines D=64 and 96 for BURSTY 1-L periodic traffic. Top line is FIFO without expired cell discarding, line with stars is FIFO with expired cell discarding, line with circles is Policy 2 and solid line is Policy 1.

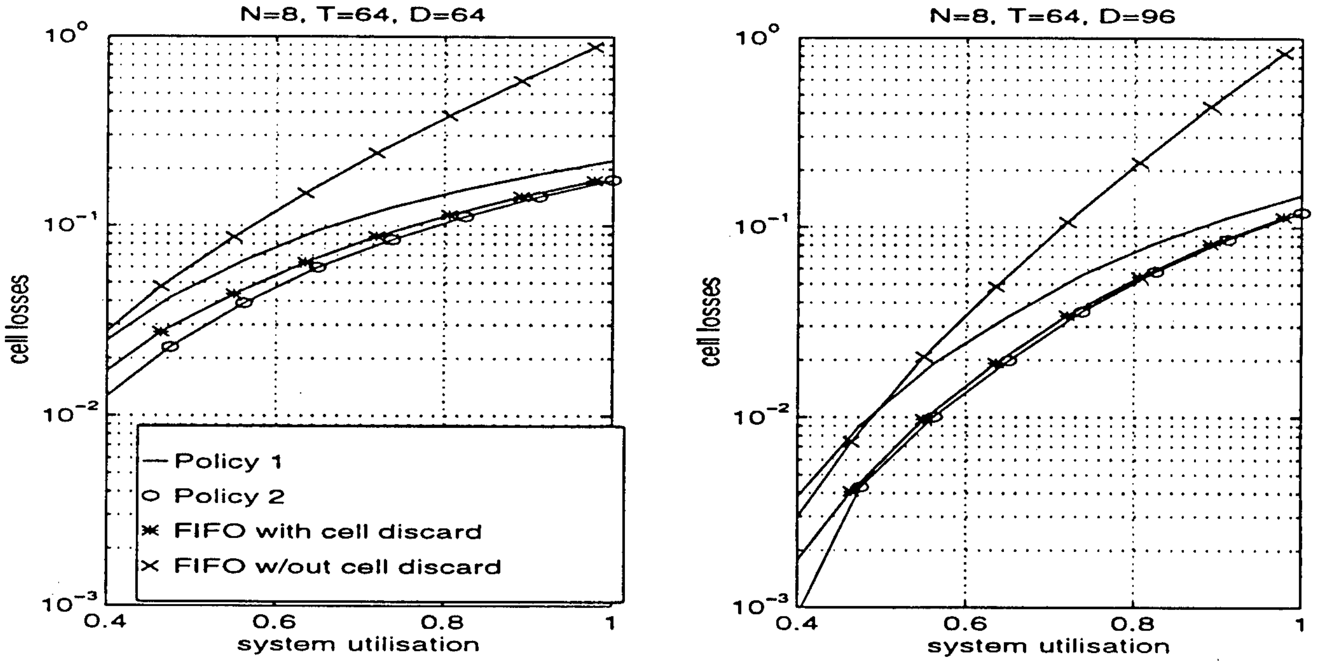


Figure 8: Cell loss curves for N=8 streams, period T=64 and deadlines D=64 and 96 for BURSTY 1-L periodic traffic. Top line is FIFO without expired cell discarding, line with stars is FIFO with expired cell discarding, line with circles is Policy 2 and bottom solid line is Policy 1.

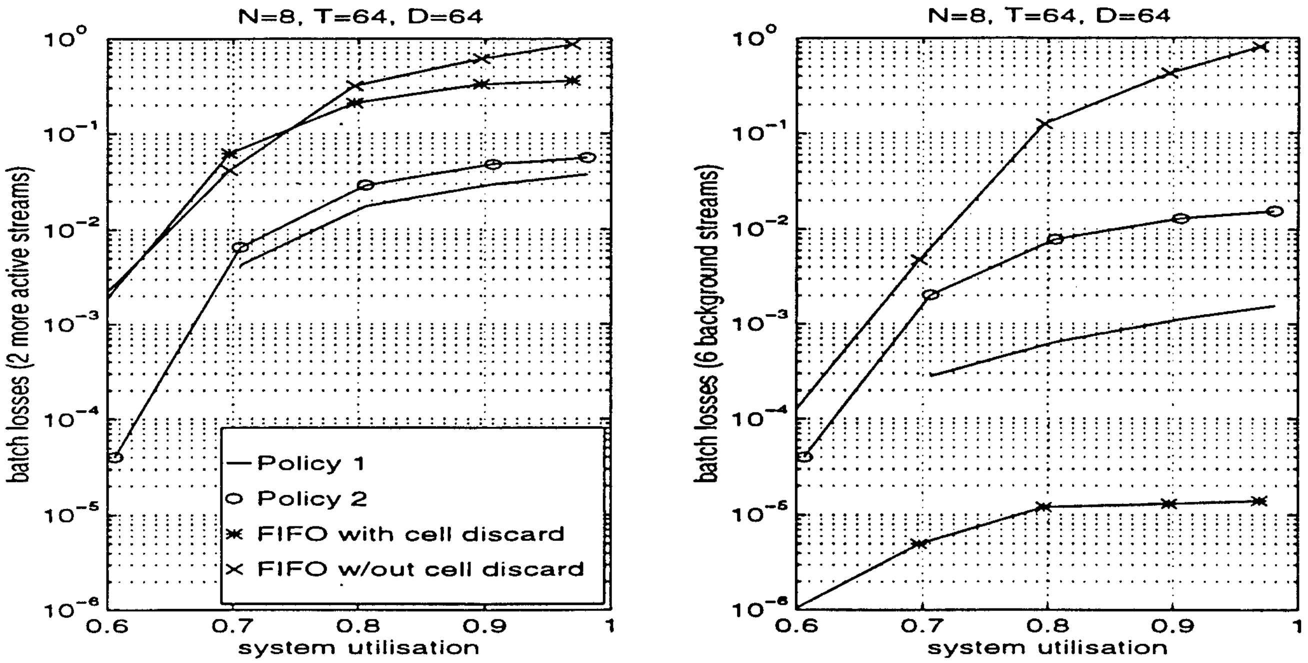


Figure 9: Batch loss curves for N=8 streams, period T=64 and deadline D=64 for INHOMOGENEOUS BURSTY 1-L periodic traffic. Top line is FIFO without expired cell discarding, line with stars is FIFO with expired cell discarding, line with circles is Policy 2 and bottom solid line is Policy 1.

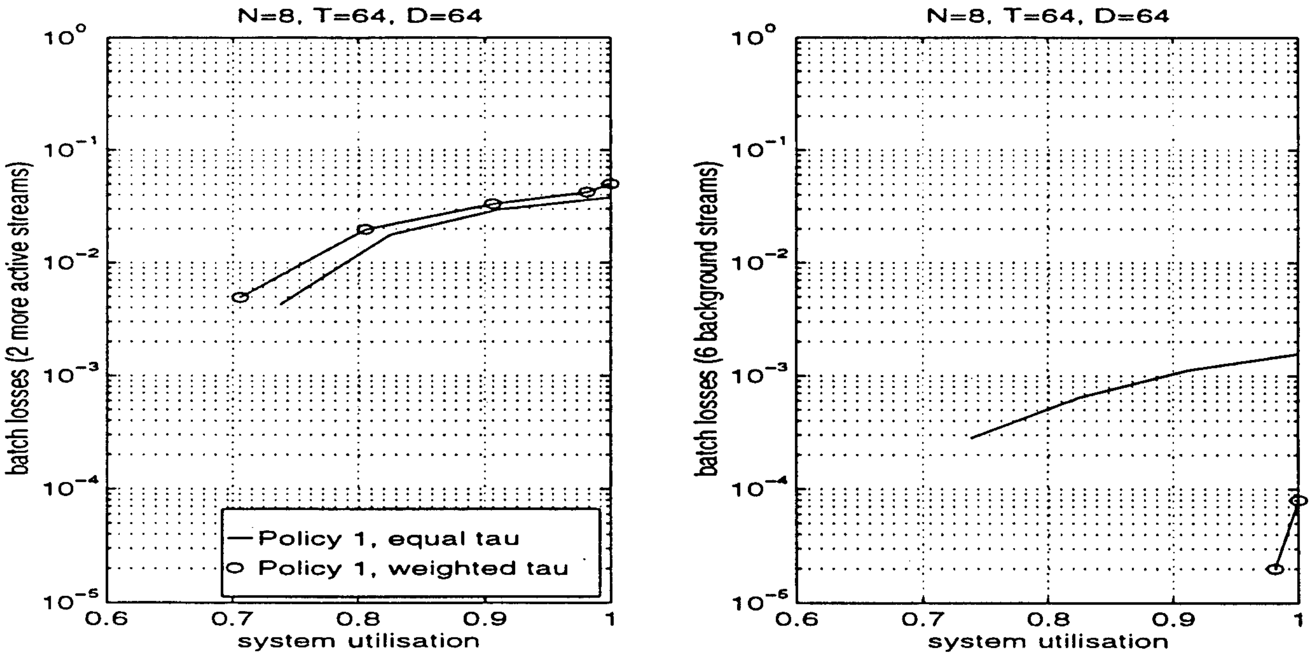


Figure 10: Batch loss curves for N=8 streams, period T=64 and deadline D=64 for INHOMOGENEOUS BURSTY 1-L periodic traffic. Solid line is Policy 1 with equal values of τ and line with circles is Policy 1 with weighted values of τ (tau).

Achievable QoS in a Shared Wireless Channel

Jeffrey M. Capone and Ioannis Stavrakakis

Communications and Digital Signal Processing (CDSP) Center
Department of Electrical and Computer Engineering
Northeastern University
Boston, MA 02115

Abstract - In this work, call admission and transmission scheduling policies are studied for a TDMA system servicing Variable Bit Rate (VBR) applications with distinct QoS requirements and traffic characteristics. In this environment, packets which experience excess delays are dropped (due to delay violations) at the source. In addition, packets are dropped at the receiver due to channel induced errors (interference) in the wireless link. The focus of the research is to determine the region of achievable QoS vectors for heterogeneous VBR applications in this shared resource environment and study the impact that channel quality has on the achievable performance. The region of achievable QoS vectors is central to the call admission problem and in this work, it is used to identify a class of scheduling policies capable of delivering any achievable performance.

I. INTRODUCTION

In *Integrated Services Wireless Networks* resources are shared among geographically disperse applications with diverse traffic characteristics and QoS requirements. The shared transmission resources are defined to be the slots (packet transmission times) of a TDMA frame. This resource structure has been widely considered in both cellular systems [1] and wireless LANs [2], as well as in recent work toward the development of wireless ATM networks [3, 4].

The focus of this work is to determine the region of achievable QoS vectors for heterogeneous VBR applications in a shared wireless environment. The QoS requirements for each application is defined in terms of a maximum tolerable packet delay and dropping probability. Determining the region of achievable QoS vectors is central to the development of a Call Admission Control mechanism. For example, if with the addition of the new source, the new multi-dimensional target QoS vector is in the region of achievable QoS vectors then the call can be admitted. If the call can not be admitted, but more resources can be made available, a precisely defined region of achievable QoS can be used to determine the minimum additional resources required in order for the new call to be admitted. In this work, the region of achievable QoS vectors is also used to identify a class of scheduling policies capable of delivering any achievable performance.

In an error-free channel all transmitted packets are successfully received. In this case, the delivered QoS is shaped by the packet discarding process at the transmitter (source) due to delay violations; the latter occur when the demand exceeds the amount of available resources for a sufficiently long period. Thus, the performance is limited by the amount of available resources (resource limited). The region of achievable QoS vectors in an error-free channel environment has been investigated in [5].

While an optical-fiber based channel may be practically considered to be error-free, this is not the case with wireless channels. Although the necessary resources may become available on time, packets may be corrupted due to channel induced errors and be dropped at the receiver. Under these conditions, the performance is limited by the interference introduced in the wireless channel (interference limited). Such packet discarding may occur with a frequency comparable to that of the packet discards at the transmitter due to resource limitations. As a consequence, the region of achievable QoS vectors is shaped by the packet discarding process at both the transmitter and the receiver due to resource and interference limitations, respectively.

II. DESCRIPTION OF THE SYSTEM MODEL

Consider a system where N VBR (Variable Bit Rate) sources compete for T slots in the up-link channel. T

Research supported in part by the Advanced Research Project Agency under Grant F49620-93-1-0564 monitored by the Air Force Office of Scientific Research (AFOSR) and the National Science Foundation under Grant NCR 9628116.

represents the number of slots in the TDMA frame available for VBR traffic. At the beginning of each frame n each source i will request a random number of slots denoted by $\lambda_i(n)$. If the aggregate demand in frame n, $\sum_{i=1}^{N} \lambda_i(n)$, exceeds the number of slots available to the VBR traffic (T) - referred to as an overloaded frame - then decisions must be made regarding the amount of service that will be provided to each source. The portion of slots under policy f allocated to source i, $a_i^f(n)$, may be less than what is required by that source, $\lambda_i(n)$, due to resource limitation. Packets from a source which do not receive service over a frame are considered to have excess delay and are dropped at the source.

The above environment could model the sharing of the up-link TDMA frame of a wireless network where request are processed at frame boundaries. Fig. 1 illustrates a typical up-link TDMA frame whose slots are allocated to various classes of services, such as Continuous Bit Rate (CBR), Available Bit Rate (ABR) and Variable Bit Rate (VBR). In this figure $T = 3$ slots are assumed to be available to the VBR applications.

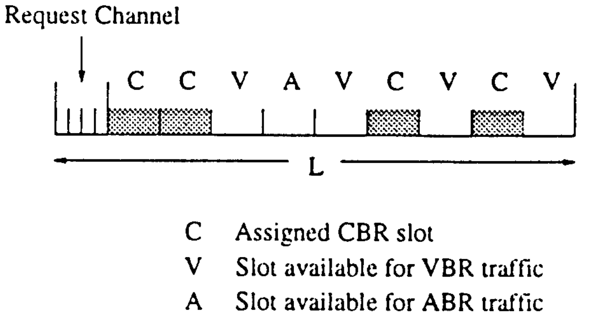

Request Channel
C C V A V C V C V
L
C Assigned CBR slot
V Slot available for VBR traffic
A Slot available for ABR traffic

Figure 1. TDMA frame supporting CBR, VBR and ABR traffic classes.

In this paper a transmitted packet will also be discarded at the receiver if the signal to interference ratio (SIR) falls below a threshold b. In a wireless channel, the received signal power may be described by taking into consideration three effects [6]: attenuation due to distance, shadowing - typically described by a log-normal distribution, and fading that causes the instantaneous envelope of the received signal to be Rayleigh distributed and consequently its power is exponentially distributed. An outage is defined as an event that occurs when the SIR falls below a predetermined threshold, b, sometimes called capture ratio. It is assumed that when a source experiences an outage its packet is corrupted, otherwise, it is correctly received. Therefore, the probability that a packet is discarded at the receiver is defined as,

$$P[SIR < b]. \tag{1}$$

Let β be an indicator function of an outage (error event). That is,

$$\beta = \begin{cases} 1 & \text{if } SIR < b \\ 0 & \text{if } SIR \geq b \end{cases}. \tag{2}$$

In this model, it is assumed that outages are independent and identically distributed; with minor modifications, correlation among outages can be easily considered.

Due to the above competition for the resources and the errors induced by the channel, the number of packets from source i dropped in frame n will depend on the policy and is given by,

$$d_i^f(n) = \tag{3}$$

$$\begin{cases} \sum_{k=1}^{\lambda_i(n)} \beta_k & \text{if } \sum_{j=1}^{N} \lambda_j(n) \leq T \\ \lambda_i(n) - a_i^f(n) + \sum_{k=1}^{a_i^f(n)} \beta_k & \text{if } \sum_{j=1}^{N} \lambda_j(n) > T \end{cases}.$$

Let $d_i^f = E\left[d_i^f(n)\right]$, $a_i^f = E\left[a_i^f(n)\right]$ and $\lambda_i = E\left[\lambda_i(n)\right]$ be the (assumed time invariant) expected values of the associated quantities.

Suppose that the QoS requirement of application i is defined in terms of a maximum tolerable average per frame packet dropping rate d_i, $0 \leq i \leq N$. Then the QoS vector associated with the supported applications can be defined in terms of the (performance) packet dropping rate vector $\mathbf{d}$,

$$\mathbf{d} = (d_1, d_2, \ldots, d_N). \tag{4}$$

When the QoS requirement of the application i is defined in terms of a maximum tolerable packet dropping probability p_i, the corresponding packet dropping rate d_i is easily determined by $d_i = \lambda_i \; p_i$.

The first question addressed in the sequel (section III) is whether (under the given channel conditions) a given QoS vectors $\mathbf{d}$ is achievable under any policy f. The second question, addressed in section IV, is concerned with the design of scheduling policies that deliver an achievable target QoS vector $\mathbf{d}$.

III. DETERMINATION OF THE REGION OF ACHIEVABLE QOS VECTORS

The establishment of the region of achievable QoS vectors is based on a set of inequalities and an equality constraint derived by employing work-conserving arguments. A work-conserving policy (denoted as f) does not waste resources (slots) as long as there is work to perform (packets to transmit). Details regarding these derivations may be found in [7].

Let $S = \{1, 2, \ldots, N\}$ be the set of all sources and $d^f_{\{S\}}$ denote the average system packet dropping rate under scheduling policy f, denoted by,

$$d^f_{\{S\}} \triangleq E\left[\sum_{i=1}^{N} d^f_i(n)\right] = \sum_{i=1}^{N} E\left[d^f_i(n)\right] = \sum_{i=1}^{N} d^f_i. \quad (5)$$

Define $\lambda_g(n) = \sum_{i \in g} \lambda_i(n)$ for any $g \subseteq S$.

Under any work-conserving, f, the average system packet dropping rate is,

$$\begin{aligned} d^f_{\{S\}} = E[\beta]\left\{E\left[\lambda_S(n) \;\middle|\; \lambda_S(n) \le T\right]\right\} & \quad (6) \\ \cdot Pr\left(\lambda_S(n) \le T\right) & \\ + \left\{E\left[\lambda_S(n) \;\middle|\; \lambda_S(n) > T\right] - T(1 - E[\beta])\right\} & \\ \cdot Pr\left(\lambda_S(n) > T\right). & \end{aligned}$$

As it can be seen from (6), $d^f_{\{S\}}$ is independent of the policy f. Therefore, the system dropping rate, $d^f_{\{S\}}$, is conserved under any policies f and is denoted as $b_{\{S\}}$.

Let $d^f_{\{g\}}$ denote the average subsystem $\{g\}$ packet dropping rate under policy f, defined by,

$$\begin{aligned} d^f_{\{g\}} \triangleq E\left[\sum_{i \in g} d^f_i(n)\right] &= \sum_{i \in g} E\left[d^f_i(n)\right] \quad (7) \\ &= \sum_{i \in g} d^f_i, \quad g \subset S. \end{aligned}$$

That is, $d^f_{\{g\}}$ is equal to the aggregate packet dropping rate associated with sources in group g only, under policy f; all N sources in S are assumed to be present and served under policy f.

Let $b_{\{g\}}$ denote the lower bound for the aggregate packet dropping rate for sources in $\{g\}$, determined to be equal to the packet dropping rate of a system in which only sources in $\{g\}$ are present and served under a work-conserving policy. Sources in set $\{S - g\}$ are considered to be removed. It is given by,

$$\begin{aligned} b_{\{g\}} = E[\beta]\left\{E\left[\lambda_g(n) \;\middle|\; \lambda_g(n) \le T\right]\right\} & \quad (8) \\ \cdot Pr\left(\lambda_g(n) \le T\right) & \\ + \left\{E\left[\lambda_g(n) \;\middle|\; \lambda_g(n) > T\right] - T(1 - E[\beta])\right\} & \\ \cdot Pr\left(\lambda_g(n) > T\right). & \end{aligned}$$

It is apparent that no policy can deliver a lower dropping rate to sources in set g than $b_{\{g\}}$, when all sources in S are present. Thus, $b_{\{g\}}$ is a lower bound on the performance induced for the set g under any policy.

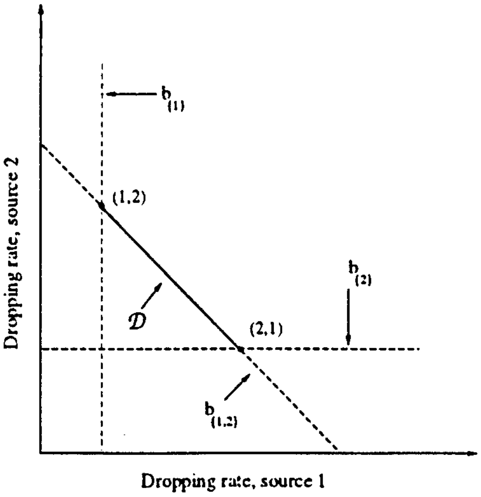


Figure 2. The region (polytope) $\mathcal{D}$ for a system with two sources.

It is shown in [7] that necessary and sufficient conditions for a QoS vector $\mathbf{d} = (d_1, d_2, \ldots, d_N)$ to be achieved by a scheduling policy f are that its components satisfy the following constraints,

$$\begin{aligned} d_{\{g\}} &\ge b_{\{g\}} \quad \forall g \subseteq S & (9) \\ d_{\{S\}} &= b_{\{S\}}. & (10) \end{aligned}$$

Let $\mathcal{D}$ denote the collection of all vectors $\mathbf{d}$ satisfying (9) and (10). Any vector in the set $\mathcal{D}$ can be expressed as a convex combination of extreme points (vertices) of $\mathcal{D}$; that is, $\mathcal{D}$ may be expressed as the convex hull of its extreme points, $\mathcal{D}$=conv[exp($\mathcal{D}$)], see [7]. It is shown in [7] that $\mathbf{d}^*$ is a vertex of the set $\mathcal{D}$ iff $\mathbf{d}^*$ is a dropping rate vector resulting from an Ordered HoL (O-HoL) priority service policy, $\pi = (\pi_1, \pi_2, \ldots, \pi_N)$; $\pi_i \in \{1, 2, \ldots, N\}$, $\pi_i \ne \pi_j$, $1 \le i, j \le N$. The index of π_i indicates the order of the priority given to the π_i source. None of the π_j sources, $j > i$, may be served as long as packets from sources π_k, $k \le i$, are present.

Fig. 2 and Fig. 3 provide a graphical illustration of the region $\mathcal{D}$ for the case of $N = 2$ and $N = 3$ sources, respectively. The extreme points correspond to QoS vectors $\mathbf{d}$ induced by the $N!$ ordered HoL priority policies $\pi = (\pi_1, \pi_2, \ldots, \pi_N)$. Referring to Fig. 2, it may be observed that the policy $(\pi_1, \pi_2) = (1, 2)$ corresponds to the intersection of the line for the lower bound on the packet dropping rate line for source 1, $b_{\{1\}}$, with the system dropping rate line $b_{\{1,2\}}$. Similarly, the second extreme point induced by the policy $(\pi_1, \pi_2) = (2, 1)$ is the intersection of the lower bound on the packet dropping rate for source 2, $b_{\{2\}}$ and $b_{\{1,2\}}$. Similar observations can be made for the region $\mathcal{D}$ for a system of $N = 3$ sources shown in Fig. 3.

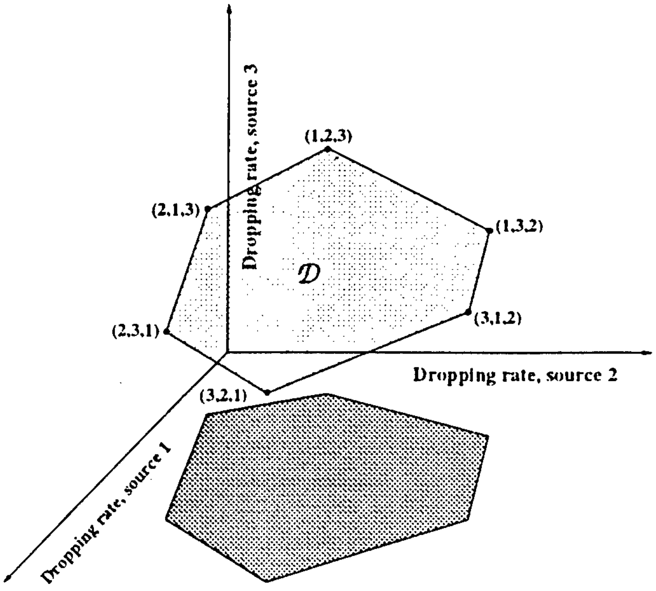


Figure 3. The region (polytope) $\mathcal{D}$ for a system with three sources.

IV. A CLASS OF POLICIES INDUCING ALL QOS VECTOR IN $\mathcal{D}$

Let C_{O-HoL} denote the class of Ordered HoL (O-HoL) priority service policies $\boldsymbol{\pi}$ introduced in section III. A mixing O-HoL priority service policy f_m is defined to be one that at each frame decides to follow the O-HoL priority policy $\boldsymbol{\pi}^i = (\pi_1^i, \pi_2^i, \ldots, \pi_N^i)$ with probability $\alpha_i, \alpha_i \geq 0$, $1 \leq i \leq N!$, $\sum_{i=1}^{N!} \alpha_i = 1$; decisions over consecutive frames are independent. Clearly, f_m is completely determined by the $N!$ dimensional vector $\boldsymbol{\alpha}$, $\boldsymbol{\alpha} \geq 0$, $\mathbf{1} \cdot \boldsymbol{\alpha} = 1$. Let M_{O-HoL} denote the class of such policies.

For each packet dropping rate vector $\mathbf{d} \in \mathcal{D}$ there exists a policy $f_m \in M_{O-HoL}$ that induces $\mathbf{d}$. The proof follows from the fact that $\mathcal{D}$ can be written as a convex combination of the extreme points (vertices) $\mathbf{d}_{ext-i}$ of $\mathcal{D}$, that is $\mathbf{d} = \sum_{i=1}^{N!} \alpha_i \mathbf{d}_{ext-i}$ for some $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_{N!})$ where $\alpha_i \geq 0$, $1 \leq i \leq N!$, $\sum_{i=1}^{N!} \alpha_i = 1$. Since each $\mathbf{d}_{ext-i}$ is induced by some policy in C_{O-HoL}, a mixing policy f_m that selects the HoL priority $\boldsymbol{\pi}^i$ (that induces $\mathbf{d}_{ext-i}$) with probability α_i such that $\alpha_i \geq 0$, $1 \leq i \leq N!$, $\sum_{i=1}^{N!} \alpha_i = 1$ will have a packet dropping rate vector $\mathbf{d}^{f_m}$ given by,

$$\mathbf{d}^{f_m} = \sum_{i=1}^{N!} \alpha_i \mathbf{d}_{ext-i} \qquad (11)$$

and thus f_m induces $\mathbf{d}$.

Let $\mathbf{d} \in \mathcal{D}$ be a target packet dropping rate vector. The mixing policy $f_m \equiv \boldsymbol{\alpha}$ induces $\mathbf{d}$, where $\boldsymbol{\alpha}$ is such that,

$$E_{\boldsymbol{\alpha}}[\mathbf{d}_{ext}] \triangleq \sum_{i=1}^{N!} \alpha_i \mathbf{d}_{ext-i} = \mathbf{d}, \qquad (12)$$

$$\boldsymbol{\alpha} \geq 0, \qquad (13)$$

$$\mathbf{1} \cdot \boldsymbol{\alpha} = 1, \qquad (14)$$

where $E_{\boldsymbol{\alpha}}[.]$ is weighted average of the set of extreme points, $\mathbf{d}_{ext}$, of $\mathcal{D}$ with respect to the Probability Mass Function $\boldsymbol{\alpha}$.

V. NUMERICAL AND SIMULATION RESULTS

In this section, numerical examples and simulations are carried out for a system with two sources competing for 5 slots in a TDMA frame. The source packet arrival processes are assumed to be mutually independent. Each arrival process is described in terms of an underlying finite-state Markov chain embedded at the frame boundaries. The number of packets generated (and requesting service) by a source in the current frame boundary is (probabilistically) determined by the present state of the associated underlying Markov chain.

Consider in this example the typical TDMA frame in which one slot per frame represents a 32 Kbps channel. Let $E^k = \{0, 1, \ldots, M^k - 1\}$ denote the state space of the Markov chain associated with source k, $k = 1, 2$. In this example $M^1 = 6$ and $M^2 = 6$. When in state i, a source generates i packets in the current frame. Source 1 and 2 could be used to model a low rate video transmissions with maximum rate 6 slots/frame = 192 Kbps. Further description of all two sources can found in Table I. The arrival rates are easily computed and are equal to $\lambda_1 = 4.2$ and $\lambda_2 = 3.2$ packets per frame.

State	State Probability	
	Source 1	Source 2
0	0	0.1
1	0.1	0
2	0.1	0.4
3	0.3	0
4	0.2	0.3
5	0.2	0
6	0.1	0.2

Table I
Description of Sources.

With the sources given above, (9) and (10) were evaluated with $T = 5$ slots and under various channel conditions. The impact channel quality has on the region of achievable QoS vectors can be seen in Table II.

The channel quality has a greater relative effect on the lower bounds for the subsets than on the system. This implies that when the system is very diversified, and some sources are requesting service close to their lower bounds, then the channel quality is an important factor in determining whether the target performance is achievable. Simulation results are presented in Table III. The previously described sources in Table I were simulated and service (with $T = 5$ slots per frame) according to various mixing policies. Each mixing policy was given by α, which is the probability of selection of O-HoL policy π=(1,2) (and therefore π=(2,1) with probability $(1-\alpha)$), and varied from 0 to 1.

Theoretical Results			
$E[\beta]$	$b_{\{1\}}$	$b_{\{2\}}$	$b_{\{1,2\}}$
0	0.1000	0.2000	2.0700
0.01	0.1350	0.2300	2.1173
0.02	0.1700	0.2600	2.1646

Table II
Impact of channel quality on the region of achievable QoS.

Simulation Results				
$E[\beta]$	α	d_1	d_2	$d_1 + d_2$
0.00	0.00	1.8835	0.1978	2.0814
	0.25	1.4323	0.6490	2.0814
	0.50	0.9837	1.0977	2.0814
	0.75	0.5358	1.5455	2.0814
	1.00	0.0969	1.9844	2.0814
0.01	0.00	1.9007	0.2279	2.1286
	0.25	1.4540	0.6746	2.1286
	0.50	1.0101	1.1184	2.1286
	0.75	0.5668	1.5618	2.1286
	1.00	0.1324	1.9962	2.1286
0.02	0.00	1.9178	0.2586	2.1764
	0.25	1.4756	0.7008	2.1764
	0.50	1.0362	1.1402	2.1764
	0.75	0.5979	1.5784	2.1764
	1.00	0.1677	2.0086	2.1764

Table III
Simulation results displaying the effects of the channel quality on the performance of various mixing policies.

Let d_k denote the target packet dropping probability, for source k, $k = 1, 2$. In this example it is assumed that $d_1 = 1.220$ and $d_2 = 0.900$. Thus, the QoS vector **d** is given by $\mathbf{d} = (d_1, d_2) = (1.220, 0.900)$ and the associated system packet dropping rate is equal to $d_{\{S\}} = d_1 + d_2 = 2.120$. By satisfying (9) and (10) (see Table II), it can be established that the performance can be achieved when $E[\beta] \leq 0.01$. When the channel condition are given by $E[\beta] = 0.02$, the system dropping rate can not be satisfied, and there the QoS can not be delivered to applications.

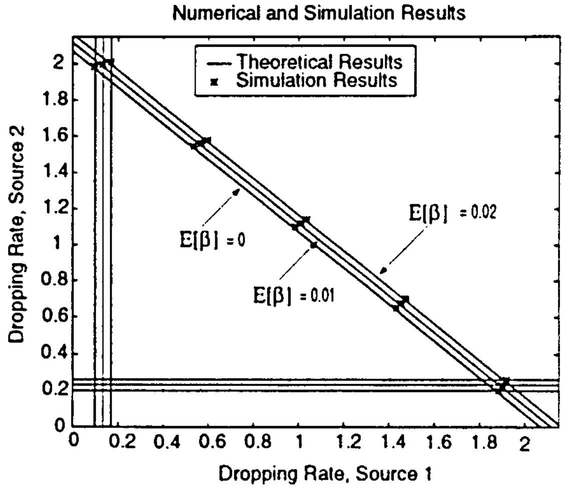


Figure 4. Numerical evaluation of the region of achievable QoS under various channel conditions for the system of sources given in Table 1 ($T = 5$) and simulation results from Table 3.

When $E[\beta] \leq 0.01$, $d_{\{S\}} \geq b_{\{S\}}$ and thus, according to [7], there exists a vector $\mathbf{d}'$ such that $d'_{\{g\}} \leq d_{\{g\}}$, $g \subset S$ and $d'_{\{S\}} = b_{\{S\}}$; $\mathbf{d}'$ is said to have improved performance. Consider the case when $E[\beta] = 0.01$, and let $\mathbf{d}' = (1.218, 0.899)$; $\mathbf{d}' \in \mathcal{D}$ and thus there exists a mixing ordered HoL priority policy $f_m \equiv \boldsymbol{\alpha}$ achieving exactly the QoS vector $\mathbf{d}'$. Any $\boldsymbol{\alpha}$ satisfying the conditions (12), (13) and (14) may be chosen. The following $\boldsymbol{\alpha}_o$ was found by employing linear programming techniques,

$$\boldsymbol{\alpha}_o = \begin{bmatrix} \alpha_1 = 0.3818 \\ \alpha_2 = 0.6181 \end{bmatrix} \tag{15}$$

In this example there exists only one solution, but in systems with greater than 2 sources more than one solution may be found. This allows for the incorporation of additional constraints representing other desirable qualities of the policies. Functions of interest may be minimized subject to the constraints presented in this paper to guarantee the delivery of the target QoS vector. For instance, among all mixing policies inducing $\mathbf{d}'$, the one which minimizes the variance of the service provided to certain sources may be identified.

A simulation was performed using the policy, $\boldsymbol{\alpha}_o$, derived for this example. The sources, as described in Table I, were simulated and generated the service demand. Resources (slots) were allocated according to the prescribed policy and the number of dropped packets per frame from each source was recorded. The simulation time was 50000 frames or 2000 seconds (i.e., 40

Source, i	Theoretical, d_i	Simulation, $\widehat{d_i}(k)$
1	1.218	1.211
2	0.899	0.893

Table IV
Comparison of time-averaged and theoretical steady-state mean packet dropping rate for each source under policy α_o.

ms frame). Throughout the simulation in each frame m, for each source i, the time-averaged mean dropping rate was calculated as,

$$\widehat{d_i}(k) = \frac{1}{k}\sum_{m=1}^{k} d_i^f(m). \tag{16}$$

The resulting time-averaged mean packet dropping rate for each source under the policy was found and is displayed in table IV. Throughout the progression of the simulation the time-averaged mean packet dropping rates were recorded and are displayed in Fig. 5.Under the policy the objective of mean packet dropping rate d_i for each source i is met, see table IV and Fig. 5.

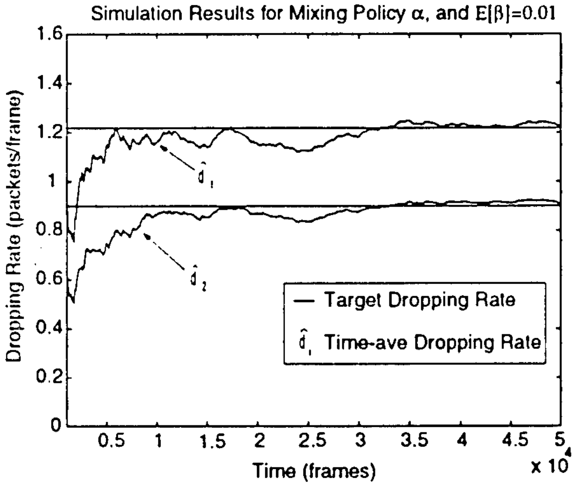


Figure 5. Simulation results obtained with the derived policy α_o.

As it was stated in section 3, satisfying the condition on the system performance, given by (10), is only necessary and not sufficient to guarantee that the target QoS vector is achievable[1]. To illustrate this concept, consider the system given in the previous example with $E[\beta] = 0$ and let $\mathbf{d} = (0.099, 1.971)$ be the target QoS vector. $\mathbf{d}$ satisfies the condition on the system dropping rate, that is $d_{\{S\}} = b_{\{S\}} = 2.07$, but $d_1 < b_1 = 0.100$. Thus, the target QoS vector cannot be achieved under any policy. For this system, the best possible performance source 1 can achieve is if it is given service priority, resulting in dropping rates of 0.100 and 1.970 for sources 1 and 2, respectively. The overall system performance is satisfied, but source 1 is experiencing poorer service than what is desired, while sources 2 is experiencing improved performance.

VI. CONCLUSION

In this work, the region of achievable QoS vectors was established for a system of heterogeneous VBR sources competing for slots of a TDMA frame. The QoS for each application was defined to be the packet dropping rate, which results from the competition for resources (slots may not come available on time and a source is forced to drop the packets with excess delay) as well as channel induced errors (corrupted packets are dropped at the receiver). Knowledge of this region was used to develop a class of scheduling policies that can deliver any achievable performance. Together, the call admission and scheduling functions can ensure the delivery of the target QoS vector under the given channel conditions.

REFERENCES

[1] F. Li and L.F. Merakos. Voice Data Integration in Digital TDMA Cellular System. In *Proceedings of International Zurich Seminar on Mobile Communications*, Zurich, Switzerland, March 1994.

[2] D. Bantz and F. Baucho. Wireless LAN Design Alternatives. *IEEE Network Magazine*, 8(2):43–45, March-April 1994.

[3] D. Raychaudhuri and N. Wilson. ATM-Based Transport Architecture for Multiservices Wireless Personal Communication Networks. *IEEE Journal of Selected Areas Communications*, 12(8):1401–1414, October 1994.

[4] M. Karol, Z. Liu, and K. Eng. An Efficient Demand-Assignment Multiple Access Protocol for Wireless Packet (ATM) Networks. *Wireless Networks*, 1(4):267–279, December 1995.

[5] J. Capone and I. Stavrakakis. Achievable QoS and scheduling Policies in Integrated Services Wireless Networks. *Performance Evaluation*, 26 and 27(1), October 1996.

[6] W.C. Jakes. *Microwave Mobile Communications*. Wiley, New York, 1974.

[7] J. Capone. Achievable QoS and scheduling Policies in Integrated Services Wireless Networks. *Ph.D. Dissertation, Northeastern University*, 1997.

[1] In the special case of a homogeneous system, such as a cellular voice system, satisfying (10) is sufficient to guarantee that the target QoS vector is achievable. This result has been established in [5].

An Integrated Regulation and Scheduling Scheme for Real-Time Traffic Management *

Steve Iatrou Ioannis Stavrakakis

Dept. of Electrical and Computer Engineering
Northeastern University
Boston, MA 02115
e-mail : ioannis@cdsp.neu.edu

Abstract

Typical rate-based traffic management schemes for real-time applications attempt to allocate resources by controlling the packet delivery to the resource arbitrator (scheduler). This control is typically based only on the characteristics of the particular (tagged) traffic stream and would fail to optimally adjust to non-nominal network conditions such as overload. In this paper, an integrated Regulation and Scheduling (dynamic-R&S) scheme is proposed whose regulation function is modulated by both the tagged stream's characteristics and some information capturing the state of the co-existing applications as provided by the scheduler. The performance of the proposed scheme – as well as that of the equivalent static one – is investigated under both underload and overload traffic conditions – both analytically and through simulation – and the substantially better throughput / jitter characteristics of the dynamic-R&S scheme are established.

1 Introduction

A secure solution to the problem of guaranteeing the QoS of real-time applications will typically require the reservation of the maximum amount of needed resources. Because of the anticipated low - due to resource requirement fluctuation - network utilization, alternative solutions are being considered based on "over allocation" of resources to a group of applications (multiplexing). Over allocation aims at improving the resource utilization, at the expense of failing to accommodate at certain times peak resource demands from a number of the grouped applications. Grouping of applications and "over-allocation" of resources are the key aspects of non-degenerate statistical multiplexing.

Because of the stringent QoS requirements of real-time applications, it is expected that "traditional" statistical multiplexing schemes,such as FCFS, will not be effective for such applications. It is well understood that some tighter control should be exercised on input output as well as in the internal processes of traffic handling. In principle, a target value of a QoS metric may be possible to achieve through either tight traffic regulation (typically referred to as rate-based approach), or sophisticated scheduling (typically referred to as scheduler-based approach) only. In most practical cases though, some scheduling will be needed to resolve transmission conflicts among rate-based controlled applications. Similarly, some traffic filtering (regulation) will be needed to eliminate extreme traffic realization which would be hard to manage even by a sophisticated scheduler under a scheduler-based approach. In addition to the supporting role of regulation in a scheduler-based approach, and scheduling in a rate-based approach, higher multiplexing gain may be achieved by allowing for some cooperation between the two functionalities which will typically co-exist and be designed to achieve a common goal. Such a dynamic policy is investigated in the present paper.

Substantial effort has been directed toward the development of regulation and scheduling schemes for real-time applications. Examples of regulation and scheduling schemes for real-time applications include : D-EDD (Delay-Earliest Due Date) [1]; J-EDD (Jitter Earliest Due Date) [2]; HRR (Hierarchical Round Robin) [3]; S&G (Stop and Go Queuing) [4]; WFQ (Weighted Fair Queuing) [5]; PGPS (Packet Generalized Processor Sharing) [6]; RCSP (Rate Controlled Static Priority) [7]; LIT (Leave In Time) [8]; MRTS (Multi Rate Traffic Shaping) [9]; VC (Virtual Clock) [10].

The traffic management scheme for real-time applications investigated in this work may be viewed as an enhancement of the Rate Controlled Static Priority (RCSP) scheme proposed in [7].

2 The Proposed Dynamic R&S Policy

The typical primary objective in regulating real-time traffic stream within the network is to control jitter or the instantaneous rate (throughput). This is achieved in the RCSP mechanism [7] by enforcing a minimum spacing

*This research is supported in part by the Advanced Research Project Agency under Grant F49620-93-1-0564 monitored by the Air Force Office of Scientific Research (AFOSR) and the National Science Foundation under Grant NCR-9628116.

at the output of the regulator associated with the traffic stream of interest (tagged traffic stream). Fig. 1 (without the feedback arrow) shows a block diagram of an architecture implementing a RCSP mechanism where each of the N multiplexed streams is regulated before it is considered for transmission.

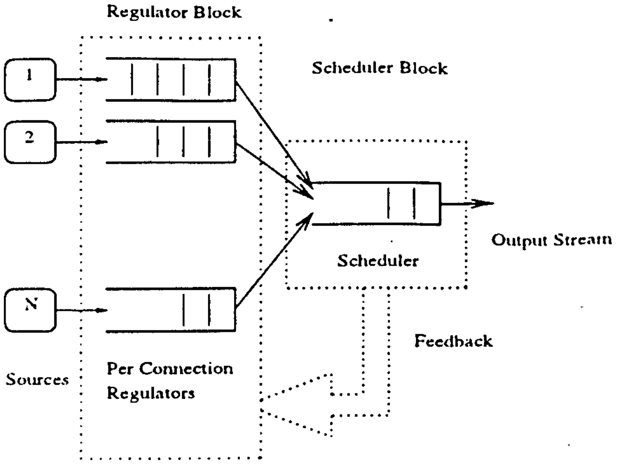


Figure 1: The Regulation and Scheduling architecture.

Since scheduling conflicts will arise when more than one regulated applications are present, a scheduler needs to be employed to resolve these conflicts. A consequence of the scheduling conflicts is that the tagged traffic stream at the output of the scheduler will be a distorted version of the target stream enforced at the output of the corresponding regulator. For instance, although a minimum spacing between consecutive tagged cells is enforced at the output of the tagged regulator in Fig. 1, this does not hold true for the tagged stream at the output of the scheduler. This clustering is generated due to an increased arrival rate to the scheduler in the immediate past which has pushed back (delayed) earlier tagged cells. Due to the latter, some spreading followed by some clustering of tagged cells is expected to be observed at the output of the scheduler.

The tagged cell spreading mentioned above can be reduced by monitoring the scheduler and releasing a tagged cell before its eligibility time [1] when scheduler queue build ups, which will cause the spreading, are detected. The dynamic Regulation and Scheduling (dynamic-R&S) scheme proposed below attempts to provide for a smoother tagged traffic at the output of the scheduler based on this idea.

Under the dynamic-R&S policy proposed below, the regulation process is modulated by some scheduler status information. Unlike in past work in the area, appropriate information regarding the status of the scheduler (FCFS queue) is fed back to the regulators, as indicated in Fig. 1 with the feedback arrow. The proposed cell release scheme is explained next.

Let t_k denote the time slot at which the k^{th} tagged cell is released from the tagged regulator. Let Q_k^r denote the queue occupancy at the regulator upon (following) the release of the k^{th} cell. Let $t_k + B_k$ ($B_k \geq 1$) denote the time slot at which the cumulative number of non-tagged arrivals (releases) to the scheduler following t_k exceeds $T - 2$ for the first time. Let a superscript d (s) indicate a quantity associated with the dynamic-R&S (static-R&S) policy and let

$$W_k = min\{B_k, T\} \text{ or } W = min\{B, T\}, \qquad (1)$$

where the last expression involves the generic random variables W and B. The $(k+1)$st tagged cell release time t_{k+1} is given by

$$t_{k+1} = t_k + W_k + H_k^d * 1_{\{Q_k^r=0, A_r^{d,W_k}=0\}}, \qquad (2)$$

where H_k^d denotes the time interval between $\overline{t_k} = t_k + W_k$ and the first tagged cell arrival following $\overline{t_k}$; T is a constant positive integer; $A_r^{d,j}$ is the number of cell arrivals to the dynamic regulator over j slots.

If T is equal to the minimum spacing among consecutive tagged cell releases from the regulator in the RCSP scheme [7], then it is easy to see that the above release policy will accelerate the tagged cell releases from the regulator at times when a minimum spacing of T at the output of the scheduler would be violated. This acceleration occurs when $B_k < T$.

The tagged cell release of the (equivalent) static R&S policy can be easily described in terms of that of the dynamic R&S by replacing W_k by T.

3 Study of Regulator Behavior

The behavior of the R&S schemes is evaluated by investigating their impact on a specific stream (tagged stream). The traffic at the output of the regulators associated with the remaining $N - 1$ applications is aggregated and forms the background traffic which competes with the tagged traffic for resources at the scheduler.

The improved performance of the dynamic-R&S policy over the static-R&S policy is established by evaluating the cell interdeparture process at the output of the scheduler. Since the input process to the scheduler is the output process from the regulator, it is important that the latter be investigated to both gain insight into the combined system (regulator plus scheduler) behavior, as well as evaluate the output process at the scheduler.

The basic operational difference between the dynamic-R&S and static-R&S schemes is captured by the tagged cell interdeparture process from the regulator $\{V_k\}_{k \geq 1}$, where $V_k = t_{k+1} - t_k$.

In order to decouple the intrinsic behavior – to be investigated in this paper – of the R&S schemes from the source load, the heavy traffic source assumption will be made throughout the paper. This assumption is consistent with standard ones made in order to determine the throughput capabilities of a scheme as well as the throughput fluctuations (jitter), without the noise introduced by

[1] Here defined as T time units following the previous tagged cell release, if a minimum spacing of T is targeted.

source inactivity periods. In view of (2) and under the heavy traffic source assumption, it is easy to establish that:

$$V_k^d = W_k \text{ (dynamic-R\&S) and } V_k^s = T \text{ (static-R\&S)} \quad (3)$$

since the indicator function in (2) is always zero. The probabilistic description of V_k^d may be found in [11] along with the detailed derivations involved in the following proposition establishing the comparative behavior of the two policies.

Proposition 1 : *The maximum throughput (output) rate of the tagged regulator under the two policies is given by*

$$R_{max}^d = \frac{1}{E\{W\}} \quad \textit{and} \quad R_{max}^s = \frac{1}{T} \qquad (4)$$

Notice that $R_{max}^d \geq R_{max}^s$ with equality only when the background traffic process can never deliver more than $T-2$ cells over $T-1$ consecutive slots.

The above establish that the dynamic-R&S scheme will respond to a sudden increase of the background load [2] by increasing its regulator's output rate above the targeted rate of $1/T$, in an effort to ensure that the targeted rate at the output of the scheduler is achieved. The impact of such a reaction (which is not possible under the static-R&S scheme) on the scheduler output process is investigated in the next section.

4 Study of Scheduler Behavior

The tagged cell interdeparture process $\{X_k\}_{k\geq 1}$ from the scheduler has been derived in order to evaluate the throughput / jitter properties of the R&S schemes. Since the deliverability of the QoS of the tagged application (which is considered to be a measure of the induced throughput / jitter) is expected to be decreased under high load conditions at the scheduler, such conditions - both within the stability and instability regions of the system - will be considered at the scheduler.

Under underload conditions ($\rho < 1$) the scheduler queue is stable and since no cell overflow is possible (assuming infinite buffer at the scheduler), the following relationships between the maximum regulator (R_{max}) and scheduler (S_{max}) throughputs hold:

$$R_{max}^d = S_{max}^d \text{ , } \quad R_{max}^s = S_{max}^s \quad \text{and} \quad S_{max}^d \geq S_{max}^s \quad (5)$$

The Probability Mass Function (PMF) of X_k has been derived under both policies by formulating appropriate queuing systems, assuming infinite buffer capacities at the regulator and scheduler and heavy traffic conditions at the regulator. The derivations - under both underload ($\rho < 1$) and overload ($\rho > 1$) - can be found in [11]. Finally it can be shown that under overload conditions at the scheduler the dynamic-R&S policy will potentially reduce the tagged cell spreading (while it will never increases it) compared to the static-R&S policy.

5 Numerical Results

The analytical results presented below have been obtained for two values of the target interdeparture time T - or desirable throughput $1/T$ - equal to $T = 5$ and $T = 10$. The background traffic is modeled as an independent per slot, batch process with binomially distributed batch size of maximum value $N - 1 = 8$ and success probability p_b. The background load or utilization is denoted by ρ_{back} (Back Util).

Figure 2 presents the regulator throughput vs ρ_{back} under both policies. As ρ_{back} increases, the dynamic-R&S policy can detect the increased background intensity and release cells earlier, attempting to provide the targeted throughput ($1/T$) and control jitter. As a consequence, the rate by which the packets leave the regulator increases as the background intensity increases. The scheduler throughput S_{max} vs ρ_{back} is shown in Fig. 3 for both policies under underload conditions ($\rho < 1$). S_{max} is calculated as $1/E\{X_k\}$. As expected from (5), $S_{max}^d = R_{max}^d$

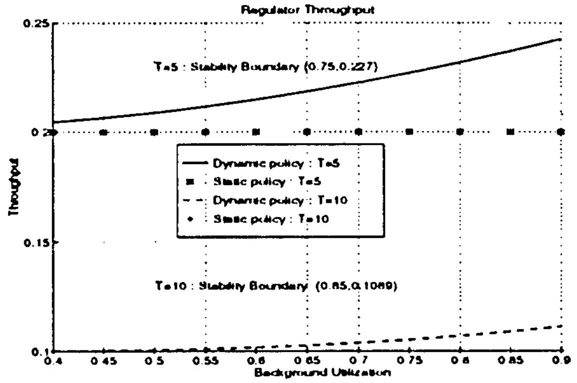


Figure 2: Regulator Throughput VS background utilization ρ_{back} ($\rho_{back} < 1$).

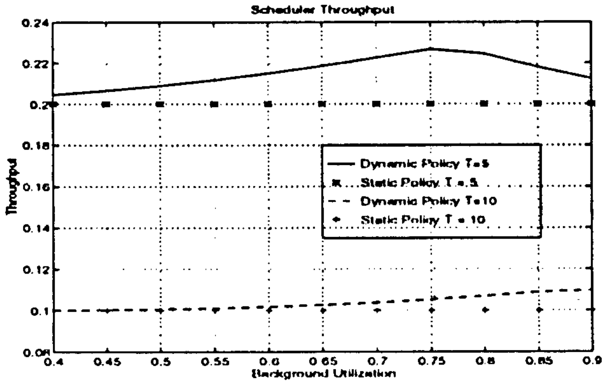


Figure 3: Scheduler throughput vs ρ_{back}

[2] which would result in an instantaneous reduction of the tagged throughput at the output of the scheduler

under underload conditions and infinite buffer capacity at the scheduler. Although R^d_{max} continues to increase as ρ_{back} increases (as said earlier), S^d_{max} starts deviating from R^d_{max} and declines beyond some value of ρ_{back} equal to about .78 for $T = 5$. This is due to the fact that the scheduler reaches an overload state ($\rho > 1$) and the dynamics change. Specifically, an infinite scheduler queue backlog is built up under these conditions, resulting in a different than R_{max} output rate from the scheduler, as discussed earlier. Thus, only the results for $R_{max} + \rho_{back} < 1$ are relevant.

The detailed traffic smoothness characteristics of the two R&S schemes are shown in Fig. 4 for underload conditions at the scheduler. Tagged cell clustering ($X_k < T$) is seen to slightly increase under the dynamic-R&S policy, while spreading ($X_k > T$) is substantially reduced and more probability mass is concentrated around T. While the spreading reduction under the dynamic-R&S policy is expected, the slight increase in the clustering is less obvious. It may be attributed to the higher probability that the scheduler queue is nonempty under the dynamic-R&S policy, due to the higher scheduler load (S_{max}) resulting from a higher regulator throughput (R_{max}).

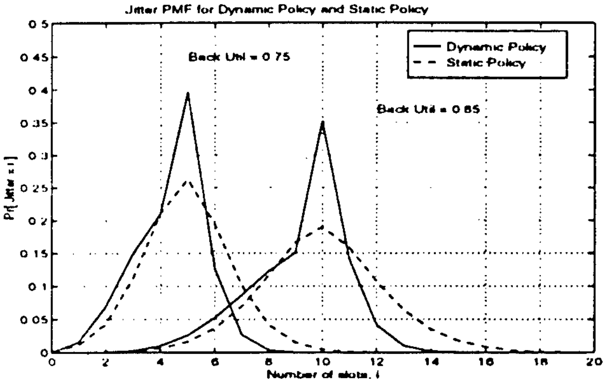


Figure 4: Jitter (X_k) PMF ($\rho_{back} < 1$)

The traffic smoothness characteristics of the two schemes under overload traffic conditions at the scheduler can be observed in Fig. 5. The jitter PMF is highly contained around the target value T under the dynamic-R&S scheme, and it is spread over wide range of values under the static-R&S scheme.

Finally, a system in which $N = 7$ ON_OFF Markov sources ($src_0 - src_6$) controlled by the R&S policies is considered and is studied through simulations;the parameters of the geometric length of the On and OFF periods are denoted by P_{on} and P_{off}, respectively. The objective of this study is to investigate whether the good characteristics of the dynamic RandS policy are maintained when multiple sources are controlled by such a policy.

The actual cell interdeparture times (X_k) from the scheduler were recorded, after filtering out the gaps caused by the source's OFF periods, Thus the calculated throughput (called the active throughput, S_a) will be higher than the actual average throughput. This filtering of the data allows for capturing the effect that the policies have on the sources while they are active, without being obscured by the OFF periods where the policies are ineffective.

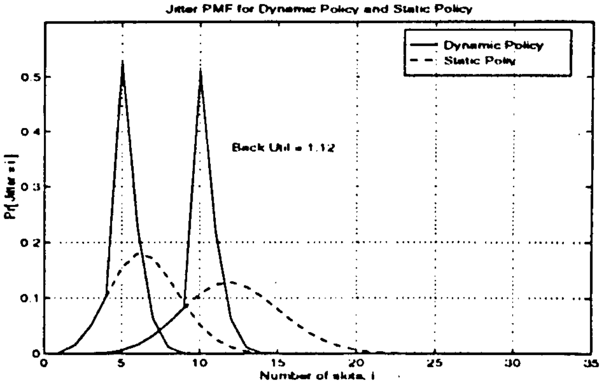


Figure 5: Jitter (X_k) PMF ($\rho_{back} > 1$)

src#	λ_{ave}	S^d_a	S^s_a	$VAR[X^d_k]$	$VAR[X^s_k]$
0	0.397	0.430	0.435	3.122	2.948
1	0.383	0.426	0.431	3.132	2.980
2	0.400	0.432	0.437	3.026	2.913
3	0.043	0.092	0.065	8.262	48.028
4	0.042	0.090	0.063	8.986	51.963
5	0.111	0.157	0.124	11.99	20.319
6	0.111	0.162	0.126	9.825	17.710

Table 1: Simulation results

Several simulation scenarios were considered with different system utilization factors. Results are shown here for $\rho = 1.49$ (overload).

The results of the simulation scenario are shown in Table 1; $P_{on} = P_{off} = .8$ for src_0, src_1, src_2, $P_{on} = .8$ and $P_{off} = .85$ for src_3, src_4 and $P_{on} = .8$ and $P_{off} = .75$ for src_5, src_6. The target cell spacings (X_{min}) enforced were: $X_{min} = 1$ for $src_0 - src_2$, $X_{min} = 10$ for $src_3 - src_4$ and $X_{min} = 5$ for $src_5 - src_6$. Sources $src_0 - src_2$ are practically unregulated since they are allowed to release cell as soon as they are generated. For each source, the average source rate λ_{ave} is shown in table 1, along with the measured active throughput S^d_a and variance $VAR[X^d_k]$ under the dynamic-R&S and S^s_a and variance $VAR[X^s_k]$ under the static-R&S policies.

The positive impact of the dynamic-R&S policy on the regulated sources ($src_3, src_4, src_5, src_6$) is clearly observed in Table 1: the variance of the interdeparture process under the dynamic-R&S scheme ($VAR[X^d_k]$) is dramatically smaller than the variance under the static-R&S scheme ($VAR[X^s_k]$). At the same time the S^d_a is higher than the S^s_a. This quantifies the ability of the dynamic-R&S scheme to control the delay jitter better than the static-R&S scheme, for all of the regulated sources. The dynamic-R&S scheme manages to serve the regulated sources with a less variable service rate and, as a result, the traffic maintains better the desirable characteristics at the output of the scheduler.

The unregulated sources (src_0, src_1, src_2) experience a slightly higher $VAR[X_k^d]$. Since the target X_{min} for these sources is 1, the two policies are basically ineffective and the variance and active throughput are shaped solely by the background interfering process seen by each of these sources during each slot. The background interfering processes are different under the two policies. It is expected that under the dynamic-R&S policy more cells will be released to the scheduler per slot during the ON period, due to the early releases.

The above results can be viewed graphically in terms of the empirical PMFs shown in Fig. 6 and Fig. 7. The PMF at the target value is substantially higher under the dynamic-R&S scheme. This may be attributed to the reduced spreading. The empirical PMFs for src_0, src_1, src_2 are almost identical under the two policies and they are not shown.

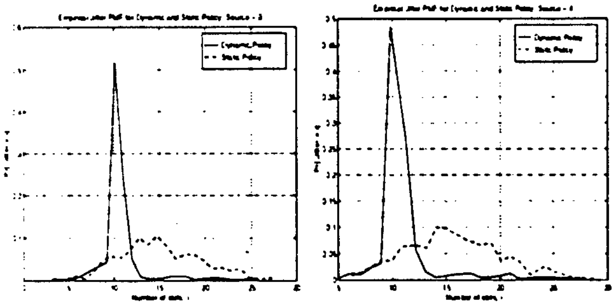


Figure 6: Empirical Jitter PMFs : src_3 and src_4

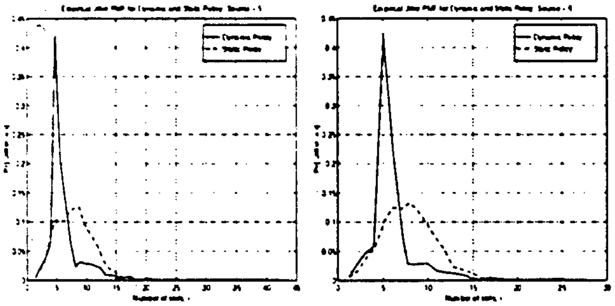


Figure 7: Empirical Jiter PMFs : src_5 and src_6

6 Conclusion

In this paper a dynamic Regulation and Scheduling (R&S) scheme has been proposed and studied through analysis and simulation. The analytical and simulation results clearly have shown that the dynamic-R&S can provide substantially better jitter control – both under underload and overload conditions – than the static-R&S scheme.

References

[1] D. Ferrari D. Verma. "A Scheme for Real Time Channel Establishment in Wide Area Networks". *IEEE Journal on Selected Areas in Communication*, 8(3):368–379, April 1990.

[2] D. Ferrari. "Delay Jitter Control Scheme for Packet-Switching Internetwork". *Computer Communications*, 15(6):367–373, July-August 1992.

[3] C. R. Kalmanek H. Kanakia S. Keshav. "Rate Controlled Servers for Very High Speed Neworks". In *IEEE Global Telecommunication Conference, San Diego California*, pages 300.3.1–300.3.9, December 1990.

[4] S. J. Golestani. "Congestion-Free Communication in High-Speed Packet Networks". *IEEE Transactions In Networking*, 39(12):1802–1812, December 1991.

[5] A. Demers S. Keshav S. Shenker. "Analysis and Simulation of a Fair Queueing Algorithm". In *Proceedings ACM SIGCOMM '89*, pages 1–12, October 1989.

[6] A. K. Parekh R. G. Galager. "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks : The Single-Node Case ". *IEEE/ACM Transactions in Networking*, 1(3):344–357, June 1993.

[7] H. Zhang D. Ferrari. "Rate-Controlled Static-Priority Queueing". In *Proceedings IEEE INFOCOM '93*, pages 227–236, September 1993.

[8] N. R. Figueira J. Pasquale. "Leave in Time: A New Service Discipline for Real Time Communications in a Packet Switching Nework". In *Proceedings ACM SIGCOMM '95 Cambridge MA*, pages 207–218, 1995.

[9] D. Saha S. Mukherjee S. K. Tripathi. "Multi-rate Traffic Shaping and End-to-End Performance Guarantees in ATM Networks". In *International Conference on Network Protocols, 1994*, pages 188–195, 1994.

[10] L. Zhang. "Virtual Clock: A New Traffic Control Algorithm for Packet Switching Networks". In *Proceedings of ACM SIGCOMM '90*, pages 19–29, September 1990.

[11] S. Iatrou I. Stavrakakis. "A Dynamic Regulation and Scheduling Scheme for Real Time Traffic Management". Technical Report TR-CDSP-97-43, Communications and Digital Signal Processing Center, ECE Dept., Northeastern University, Boston, Ma, March 1997. (also at `http://www.cdsp.neu.edu`).